ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 14 (2024) No. 1, pp. 203-227
DOI:10.33168/JSMS.2024.0113

Model for Dynamic Business Process Integration and Visualization Using Business Intelligence and SOA

Hisham Abdullah[1, *], Azman Taa[1], Fathey Mohammed[2], Mohamad Sabri Sinal[1]

[1] School of Computing, Universiti Utara Malaysia (UUM), 06010 Sintok, Kedah Darul Aman, Malaysia

[2] Department of Business Analytics, Sunway Business School, Sunway University, Selangor, Malaysia

h.alruaini@gmail.com (Corresponding author)

Abstract. Integrating Business Processes (BPs) from diverse software systems poses challenges in adapting to changing requirements, modifying process execution flow at runtime, and maintaining data interoperability while visualizing the workflow. The complexity arises from using different tools and languages for each model or phase, leading to costly transitions between models. Additionally, the separation between process identification and design hinders the identification of reusable functionalities and data interoperability. To address these challenges, this paper presents a dynamic model for integrating BPs and visualizing their workflow using Service-Oriented Architecture (SOA). The proposed model comprises five components: interaction, service, process, integration, and intelligence. A real case study is employed to demonstrate the model's effectiveness, and expert evaluations are conducted for its assessment. The results showcase the model's efficiency in enabling dynamic BP integration, accommodating changes in business process designs without extensive system redesign, and facilitating real-time monitoring support. The proposed model empowers adaptable and visualized processes that can be modified at runtime, streamlining the integration process effectively.

Keywords: Service Oriented Architecture (SOA), Business Processes (BPs), Process Integration, Process Flow Visualization, Disparate Software Systems

1. Introduction

Today, organizations of all sizes need dynamic integration solution to streamline processes for increasing productivity by simplifying regular enterprise functions (Fajar & Legowo, 2018). However, the complexity of a business environment frequently makes organizations produce multiple and inconsistent perspectives on the same business process, resulting in fragmentation and inefficiencies (Belchior et al., 2020). Integrating processes helps eliminate duplication and streamline workflows, leading to improved efficiency, and ensures data interoperability between processes. Moreover, with integrated processes, organizations can access real-time data and make informed decisions, and their ability to quickly adapt to changing business needs is enhanced as well. On the other hand, existing software systems are mission-critical, embedding hidden knowledge that is still of significant value and they execute effectively and accurately critical and complex business logic. Thus, they cannot be replaced (Abdellatif et al., 2021).

Business Process Integration (BPI) defines a business process model that describes the activities and tasks and coordinates the execution of business processes that span multiple disparate software systems. However, there are several integration requirements that should be considered during the integration of business processes implemented in existing software systems and applications, such as data interoperability, dynamic process flow, Business Process (BP) identification, and avoiding overlapping and duplication of the integrated BPs (Paniagua, Eliasson, & Delsing, 2019; Matejaš & Fertalj, 2019; Daraghmi, Zhang, & Yuan, 2022).

In addition, Business Intelligence (BI) has evolved from a data-oriented to a process-oriented, allowing for business process visualization based on real-time and actionable data (Ali & Mohamed, 2018). Therefore, BP visualization and monitoring the execution of BPs in real-time should be considered when integrating the BPs that are implemented in existing software systems and applications (Zafary, 2020; Abai et al., 2017). Service-Oriented Architecture (SOA) is a design approach that allows developers to develop integration modules for integrating business processes. SOA includes a set of guidelines and techniques for creating interoperable services as the basis for system integration solutions (Mircea, 2012). To integrate business processes that span two or more heterogeneous systems and applications, the Enterprise Service Bus (ESB) must leverage complex process integration as well as service mediation. BPI requires service orchestration, business process state management, the design and control of complex business logic, long-running transactions, and application integration.

A lack of adequate BPI and visualization models makes system implementation complex, increases resource consumption, and results in greater changes to the infrastructure of existing systems and applications. Previous integration models attempted to integrate existing systems and applications using the expertise of integration practitioners, with little attention paid to analyzing business requirements and orchestrating complex BPs (Mateja & Fertalj, 2019; Kaburuan et al., 2019). Other previous integration models were general frameworks that attempted to meet the needs of a diverse range of existing systems without producing concrete results (Kähkönen, 2017; Raminder & Sangeeta, 2018; Srimathi & Krishnamoorthy, 2019; AlHawari et al., 2017, Ovum, 2017). As a result, process-oriented BI combines BPM and BI to integrate the existing functionalities and activities of existing systems within BPs (e.g., decisional processes) via decision services and then tie BPs to business rules that can be changed at any time.

The existing literature revealed that there is a necessity for developing a new BP integration model tailored to the concrete business requirements to address the issues related to dynamic BP integration, such as data interoperability, process identification, service and process overlapping and duplication, fixed process flow, complexity and systems’ redesign, and process visualization (Kähkönen, 2017; Raminder & Sangeeta, 2018; Abdellatif et al., 2021; Fajar & Legowo, 2018). To overcome these problems, the identification phase should be considered the first phase in the BP integration lifecycle, and the separation between all phases of the BP integration should be reduced by developing shared

SOA services and BP representations that allow for integrating, visualizing, coordinating the distributed transactions that span multiple SOA services, and automatically synchronizing both the BP model and its instances.

The goal of this paper is to develop a model for integrating the business processes of existing disparate software systems and visualizing the data obtained from the execution of the process flow diagrams. The developed model is referred to as the Service-Oriented Process Integration and Intelligence (SO2PI) Model. SO2PI extends the capabilities of integrating BPs to be used not only for integrating BPs but also for: (1) identifying BPs based on the functionalities of existing systems; (2) directly utilizing the BP models for integrating the BPs that span two or more software systems; (3) creating instance objects that implement the SOA services and capture the operational data of business processes at runtime; (4) monitoring business processes instances and visualizing their workflows in real-time; and (5) maintaining data interoperability (Correia et al., 2021). As opposed to previous models, the proposed model does not require significant redesign in existing systems when business processes designs change, nor does it require additional adaptations to support real-time monitoring. As a result, it facilitates dynamic integration for internal and external business processes, where the flow of integrated and visualized processes can be changed at runtime. In addition, the problematic separation of across BPI phases is avoid which can contribute to the flexibility and reusability of both the SOA service and business processes and enable a truly dynamic BPI in which flow of business processes can be changed at runtime without affecting the runtime code. In the context of this study, the case study is implemented in Higher Education Institution using the following existing software systems: Student Information Systems (SIS), Finance Management System (FMS), Human Resource Management System (HRMS), and Learning Management System (LMS).

2. Literature Review

2.1. Business Process Integration based on SOA

The BPI model based on SOA includes five major components: interaction, service, process, integration, and intelligence (Butt, 2020; Xu et al., 2019). In addition, BPI models include four phases: identification, design, configuration, and implementation (Xu et al., 2019). During the identification phase, the processes should be identified based on the interactions between existing software systems. The service and process identification aims at identifying services (e.g., the reusable SOA services that are candidates for integration) and processes (e.g., the business processes that span across two or more software systems) to be integrated, executed, monitored, and controlled. In the design phase, the identified integration and visualization Application Programming Interface (API) services and integration architecture should be designed, and the candidate integration processes should be created and validated. In the configuration phase, the link between SOA services and the activities of the business process model is configured. Finally, the designed and configured SOA services and business processes are implemented, and the operational business process instances are integrated, executed, and controlled using the integration module in the implementation phase.

Accordingly, information on the execution of operational business processes and SOA services should be recorded and visualized in order to identify potential enhancements and required adaptations for enabling dynamic BPI (Abdullah, Taa, & Mohammed, 2021). Specifically, integrating and visualizing the business processes that are managed by existing systems using SOA requires mapping the Business Process Model and Notation (BPMN) model into an executable representation in the Business Process Execution Language (BPEL), as well as connecting activities of business process to integration services. This is ideal when the functionalities of existing software systems that match the activities of disconnected business processes are identified and the reusable SOA services that implement these functionalities are developed (Mateja & Fertalj, 2019). Hard-coding, metadata annotations, and controlled vocabulary are data interoperability techniques that were adopted previously to address semantic interoperability (Belete, Voinov, & Laniak, 2017).

According to Goldstein et al. (2019), the separation of design-time and runtime (i.e., the implementation phase) isolates the runtime process instances from directly utilizing the BP model defined in the design phase. The authors justified that any adaptation, in order to react to new business requirements, would need to modify the original (build-time) BP model correspondingly. However, the separation between identification-time and design-time was not considered, as well as the isolation of implementation of existing systems from BP models, which still represents a gap for providing dynamic BPs integration and visualizing their workflow at runtime (Gonzalez-Lopez and Bustos, 2019). Contrary to the traditional approach, where Business Process Management Systems (BPMS) are used only in the initial stages of integration development, it is critical to directly utilize the BP models to effectively control and manage the execution of BPs together with their integration flows across the existing software systems. For example, rather than defining service orchestrations as fixed flows to support BPs, it is critical to define API services and BPs' activities using pre- and post-conditions to integrate BPs based on SOA and consume the services exposed as RESTful APIs. Another challenge in supporting dynamic BP integration at runtime is that adaptation of BP integration is not only dependent on the BP model but also on dynamic SOA services.

For the development of API services, the REST protocol is used, and resources are made in the JavaScript Object Notation (JSON) format. SOA enables orchestrating business services, acting as a mediator between service providers, and exposing the services to be used in a variety of processes. The combination of the two concepts of SOA and BI enables the BPs to benefit from the semantic business model provided by BI tools. The BPs are then modeled to be integrated into existing software systems as reusable integration services using SOA and RESTful WS (Fajar, Nurcahyo, & Sriartnasari, 2018; AbouEl-Seoud, AboGamie, & Salama, 2017; Borse et al., 2019; Kintz, Kochanowski, & Koetter, 2017).

2.2. Business Intelligence (BI) and Business Process Integration

The process-oriented BI supports the business process optimization based on real-time and actionable data (Talaoui & Kohtamäki, 2020). Previous BI research can be classified as ranging from data-oriented BI to process-oriented BI. Most of the BI research has focused on data-oriented BI, which is concerned with integrating data from multiple data sources into a single data source without considering the BPs that are implemented by operational systems. According to Ray and Kumar (2019), the main purpose of integrating BI to ERP is to identify and implement the critical business processes to meet the customer demands to a large extent. By reviewing the previous research in the field of process-oriented BI, most of them focused on integrating the BI with the business processes that are managed by the commercial Enterprise Resource Planning (ERP) (Santoso, 2018; Vinaja, 2018; Ray & Kumar, 2019; Aldossari & Mokhtar, 2020). Researchers justified their proposed frameworks by claiming that the BPs managed by commercial ERP are deficient and not sufficient to meet the market needs through providing the operational BI at operational level. Santos et al., (2020) discussed how to apply BI to business processes, which are managed by the commercial ERP. There is no clear standard for identifying the critical operational business processes that need to be integrated and visualized using BI. As a result, stakeholder interviews are the better method for identifying critical business processes. The author, on the other hand, proposed the business processes (To-Be) modeling, which is based on a cross-functional flowchart. Talaoui and Kohtamäki (2020) supported the argument of this research that there is a lack of practical research for integrating BI with business processes in order to improve and optimize the business processes by providing real-time information and actionable data about BP execution.

In summary, business processes that are managed in existing systems have some issues, such as redundant and duplication of tasks, fixed process flow, a lack of process identification, data interoperability, overlapping and duplication of BPs, complexity, and invisible BPs. These issues make BPs inefficient, raising the costs incurred by organizations (Kaula, 2020; Brian, Muramuzi, & Kanyunyuzi, 2019; Kopp & Orlovskyi, 2019; Russman, Seymour, & Belle, 2017; Zafary, 2020; Trieu, 2017; Linthicum, 2017; Somya, Manongga & Pakereng, 2018). Business Intelligence (BI) techniques should be applied to BPs to support dynamic BP changes and improve the flexibility and adaptability of BP modeling.

2.3. Business Process Visualization and Monitoring

As for BPI and workflow visualization, the limited research reported in the literature are confined to areas such as integrated frameworks/models based on SOA and Business Process Model (BPM) (Fajar & Legowo, 2018; Kaburuan et al., 2019; Serrano, Pérez & Alarcón, 2014; Somya, Manongga, & Pakereng, 2018; Jakimoski, 2016; Matejaš & Fertalj, 2019). For example, the framework adopted by Kaburuan et al., (2019) does not provide a sufficiently detailed discussion of the conceptual model presented. In addition, it doesn't identify the design principles, the role of its components or the technologies, and the guidelines for its implementation. The previous frameworks that were adopted for integrating the existing systems are still lacking for business use cases or a real-life problem along with some kind of evaluation. Some previous integration models focused primarily on replacing or redesigning the existing systems and applications (Kaburuan et al., 2019; Dell Boomi, 2018; Shi & Wang, 2018). Implementation on a business use case, or a real-life problem that inspired the creation of the model, along with some kinds of evaluation are also lacking. BP Integration (BPI) allows for automation of BPs, integration of systems and services, and sharing the data across numerous applications (Ma, & Molnár, 2019). Traditional BP modeling requires extensive documentation and is incapable of meeting dynamic requirement changes. Integrating BPs in existing environments, where BPs are managed by existing systems and applications is a challenging task (Matejaš & Fertalj, 2019). The operational BPs are implemented by their existing systems and applications, which may be already integrated or still heterogeneous (Estefania, et al., 2018, Mateja & Fertalj, 2019). However, for some of the BPs of existing software systems that have already integrated, there is still a semantic gap between BPs and actual existing system integration because the integration between these systems was designed using User-defined Functions (UDF), static flows of BPs with predefined mapping parameters, and fixed pre and post conditions (Jakimoski, 2016; Hrabala, Opletalova, & Tomas, 2017; Fajar, Nurcahyo, & Sriartnasari, 2018; Ray & Kumar, 2019). Therefore, the semantic gap as well as the overlapping and duplicating of integration processes makes it very difficult for the organizations to respond to the business changes or improve their BPs to meet the market demands (Javidroozi et al., 2019). BP's interoperability has become a major challenge for integrating systems with different architecture, protocols, and semantics (Paniagua, Eliasson, & Delsing, 2019). Several earlier BPI frameworks emphasize integrating the business process performed by multiple organization partners. In other words, an integrated enterprise system is inevitably interoperable (Chalmeta and Pazos, 2015). However, the combination of BP, BI, and existing systems' integration should be accomplished within a single organization. Thus, the flow of information between the departments and integration of business processes both are significant for the integrated system, which can lead to inter-operation and inter-coordination as part of an integrated enterprise (Javidroozi et al., 2019). Moreover, it is critical to avoid overlapping and duplicating integration and optimization services, and maximize integration service reuse (Brian, Muramuzi, & Kanyunyuzi, 2019; Kähkönen, 2017; Kaburuan et al., 2019). The BP integration does not completely address the issues of improving BPs, which implemented in existing systems (Santoso, 2018; Vinaja, 2018; Ray & Kumar, 2019; Aldossari & Mokhtar, 2020). However, continuous improvement requires applying BI techniques to the BPs as part of the BPI framework. Real-time BI is also required for BP monitoring the BP execution and supporting decision-making about BP improvement and efficiency.

While all the research mentioned previously attempted to address some issues, it did not address all issues related to the dynamic BP integration, such as data interoperability, process identification, service and process overlapping and duplication, fixed process flow, adaptability and effectiveness, monitoring and real-time capability, complexity and systems' redesign, and process visualization. However, analyzing the previous classic models for integrating BPs into environment of existing systems revealed that they are primarily focused on the redesign of existing systems, with minimal use of existing functionalities and reliance on existing structures. Therefore, a novel BP framework for integrating BPs into existing systems and enabling dynamic flow for the integrated BP is still required. According to (Brian, Muramuzi, & Kanyunyuzi, 2019), some BPs are integrated based on static or unmodifiable process logic. In addition, [5] argued that the static integration solutions make it difficult to change the order of business process execution. Some organizations define the metadata for integration at the technical-interface level, and any changes to these metadata requires modifying the source code of the related applications. The metadata for the integration, on the other hand, should be defined at the BP level rather than the technical-interface level.

3. Service-Oriented Process Integration and Intelligence (SO2PI) Model

Literature reviews, expert surveys, and content analysis of related approaches, frameworks, components, concepts, design principles and requirements were conducted to identify requirements, design principles, layers, and components of the SO2PI model for service-oriented business process integration and intelligence. The requirements, design principles, layers, and components of the BPI such as service-orientation, real-time capability, service modularity, transactions coordination, process automation, process modeling, and integration, workflow integration, real-time monitoring, operational BI dashboards including Key Performance Indicators (KPI), and process simulation for process optimization were considered. Business process interoperability is important for coordination and orchestration of constituent systems. The technologies of SOA, BPM, and BI were integrated with business process integration techniques to come up with a theoretical model of developing the proposed SO2PI Model. In addition, comparative analysis of existing models, approaches, and frameworks of relevant Business Process Integration (BPI) models were carried out to identify critical integration process, components, and layers of BPI models and frameworks. With the information gathered and theoretical model guides, the intended model was developed. The model was then deployed to create a prototype of the real SO2PI development, implementation, and support in order to validate its usability, quality, adaptability, and effectiveness and the model's ability to address the research problem. These were iterative processes in informal experimental form, which later were evaluated and confirmed by expert's review. The prototyped modules were developed and then systematically tested for its usability, quality, scalability, flexibility, adaptability, and effectiveness and ability to tackle the research problem in two real case studies of actual implementations of BP integration and intelligence.

3.1. SO2PI Requirements

The review of related works revealed that the model of BPI should avoid extensive redesigning in existing systems and replacing the architecture of existing heterogeneous systems and applications. In addition, it should support real-time BI for monitoring the execution of business processes that span across multiple software systems. The proposed SO2PI model should be adaptable to achieve a loose coupling between a BPM and the existing systems. In addition, SO2PI should meet business changes and allow dynamic business process flow by modifying the business logic contained in the SoBPFA. The flow of integration process must be dynamic. Moreover, changes in processes must be automatically visible in the underlying software systems and applications. To ensure flexibility, the BPI model must be able to interact with different BPM tools simply by adapting the implementation of its APIs for the new tool. Finally, the BPI model needs to support the dynamic process flow, which would make it easier to integrate and use various heterogeneous systems and applications developed in various technologies and from various parts of the organization in a BPM solution at runtime.

This section defines Integration Requirements (IR) that a proposed integration model should satisfy considering the BP integration components, design principles, and phases discussed in the introduction and the literature review presented in Section 2.

1) *IR1 (Data interoperability):* The SO2PI should be able to maintain data and interoperability when moving the data among disparate software systems through the tasks of BPs that are candidates

for integration. *Rationale*: The integration module should be able to overcome the data transformation among the activities of BPs. It is often necessary to perform additional computations on data as it is transferred from one BP of software system to another.

2) *IR2 (Process identification):* The SO2PI should be able to identify the processes that need to be integrated. Then it should be able to identify tasks of integrated BPs in the BP model that represent reusable functionalities in existing software systems. There should be a transparent transition from the identification phase to the design phase without the need to generate or implement additional integration code. *Rationale*: The integration module should directly expose the functionalities of existing systems. In addition, the business analysts need to identify the BPs need to be integrated and they should eliminate the repeated and redundant tasks. In addition, integration specialists need to know how each task of integrated BP between two different software systems is linked to API service. As a result, the two different software systems can be immediately integrated once the design of the integrated BP between these two systems is completed, and the SOA services are linked to the tasks of BP.

3) *IR3 (Dynamic process flow):* The proposed SO2PI model should be able to directly design and utilize the BPs that span multiple software systems. That is, there should be a seamless transition from the BP design to the BP integration without the need for additional integration code to be generated or implemented. As a result, BPs can be immediately executed once the design of the BP is complete. *Rationale*: The integration module should directly read the flow of integrated BPs from the BP model. As a result, the flow of integrated BPs can be changed and modified in the BP model at runtime without affecting the implementation of integration. In addition, the business analysts and integration specialists need to know how each BP is defined to be integrated between two different software systems.

4) *IR4 (Adaptability and effectiveness, reduce complexity and systems redesign)*: The SO2PI model should be easily adaptable and flexible to changes in BP designs without the need to modify or add the integration code. *Rationale*: The SO2PI model should avoid extensive redesigning in existing software systems and replacing the architecture of these systems. In addition, the SO2PI model should be on adaptability to achieve a loose coupling between a BPM and the existing systems. Due to BPM tool failures or upgrades, without a BPM, existing systems must remain operational, and their BPs are internally integrated. Thus, business analysts and integration specialists need to modify integrated BP to meet changing needs without making significant redesign in existing systems. Consider whether the new task should be added to the integrated BP. Only the new task should be linked to the SOA service at runtime, with no need for integration to be reimplemented.

5) *IR5 (Monitoring and Real-time Capability):* The SO2PI model should support real-time BI for monitoring the execution of integrated BPs in real-time (e.g., current state of integrated BPs' execution, completed BP activities and tasks, etc.) and analyze the performance of integrated BPs (e.g., durations and costs of different tasks and probabilities of various events) at runtime (Hasic, Vanwijck, & Vanthienen, 2017; Zhao, 2017). The transition from the implementation phase to the intelligence phase should be dynamic and transparent. Integrated BPs should be directly ready for real-time monitoring and analysis; that is, no additional pre-processing or preparations, such as data extraction for different data source, cleansing, or data type conversions of execution history, should be required to monitor the executions of integrated BPs and analyze the results of their executions. *Rationale*: Business analysts and integration specialists need to know when a particular process, which spans two or more systems, started, and terminated, what the average execution time of all integrated business processes of a certain type in a specific year was, or what the response value of an API service call was.

6) *IR6 (Visualization):* The SO2PI should support visualizations for representing the workflow of integrated BPs in the design, and monitoring phases. *Rationale*: Using similar visualization for different tasks of integrated BPs would improve communication between the different software systems.

7) *IR7 (Reduce Overlapping and duplication):* The SO2PI should reduce the service and process overlapping and duplication. *Rationale*: many processes need to be grouped and categorized and controlled to maximize the reuse of service and process.

3.2. SO2PI Development

The essential technologies, along with their design principles, core components and elements, are identified and then refined for developing the SO2PI model. Seven integration requirements are established for the BP integration model that supports the five components: interaction, service, process, integration, and intelligence. Table 1 shows how each component of the proposed SO2PI model is used to address one or more of integration requirements.

Table 1. Linking the integration requirements to the components and phases of the proposed SO2PI model

Integration Requirements	The components of SO2PI Model	Major Component
IR1, IR4	▪ Service-oriented Integration and Coordination Adapter (SoICA) ▪ Service-oriented Business Process Communication Layer (SoBPCL)	Integration Process Service
IR2, IR7, IR4	▪ Business Process Identification Layer (BPIL) ▪ So2PI Service Layer	Interaction Process Service
IR3, IR6, IR4	▪ Service-oriented Business Process Flow Application (SoBPFA)	Process
IR5	▪ Service-oriented Real Time BI (SoRTBI) ▪ Service-oriented Business Process Communication Layer (SoBPCL)	Intelligence Integration Process

The requirements of the SO2PI were specifically derived from the existing literature and feedback from software industry consultants, professionals, and researchers to ensure consensus with respect to the requirements of dynamic BPI. Table 2 shows how some SO2PI model components and design principles were inspired by previous models' components.

Table 2. Some SO2PI model components which inspired by previous models' components.

Framework Component	Sub-components	Author/Year	Proposed Customization
Business Process Communication Layer	Business process communication	*Journal*: Matejaš & Fertalj (2019)	Adding the following two steps to increase performance and avoid BP duplication and overlapping: • Separating the internal BP integration from external BP integration • Separating decisional processes from integration processes
Enterprise Service Bus layer	Internal communication layer	*Journal*: Kaburuan, JM, Suprapto, & Januar(2019)	• Developing new software module by which the BPs and SOA services are orchestrated as well as the distributed transactions are coordinated.
Business Layer	Business Process (BP) Model		• Grouping the BPs into three groups: analysis processes, integration processes, and decisional processes
Workflow Management Layer	Process workflows	*Journal*: Al-Barakati (2021).	• Developing new software application by which the BPs are executed based their flows
Business intelligence	Business rules. Business Intelligence Engine.	*Journal*: Al-Barakati (2021). Journal: Wang & Lee (2011)	• Customizing the Business rules for supporting all three groups of BPs • Visualizing the BP analysis based on the results of executing the BPs

Firstly, the Business Process Communication Layer was inspired from the model developed by Matejaš and Fertalj (2019). However, some customizations were added to the proposed SO2PI model to increase performance, avoid BP duplication, and overlapping as follow: separating the internal BP integration from external BP integration as well as separating decisional processes from integration processes. Secondly, the ESB Layer was inspired from the framework developed by Kaburuan, JM, Suprapto, and Januar(2019). However, some additional customization such as developing new Service-oriented Integration and Coordination Application (SoICA) module to be implemented in ESB. The SoICA was developed to orchestrate the BPs and SOA services as well as coordinate the distributed transactions that span multiple SOA services. The customization in this step was added to the proposed SO2PI model for maintaining the data interoperability. Thirdly, Workflow Management Layer and Business Intelligence Engine were inspired from the frameworks developed by Al-Barakati (2021) and Wang & Lee (2011) respectively. However, some additional customization such as developing new Service-oriented Business Process Flow Application (SoBPFA) and applying the BI techniques to the SoBPFA. This enables the organization to meet business changes and allows dynamic BP process flow by modifying the business logic contained in the proposed SoBPFA application. In addition, it supports the BPs dynamic integration and optimization without significant redesign and support dynamic BP control and coordination. The customization in this step further provides real-time BI for the BPs analysis.

In addition to the previous customizations, the SO2PI includes the following features:

- Business Process Identification Layer (BPIL) is proposed to identify the PBs that are candidate for integration, decisional, and analysis. It forwards the BPs to SoBPFA to model the flow of these BPs. In addition, the BPIL is used to group the BPs under their related business domains for facilitating management and reducing the complexity of the BPs. It separates the BPs that are candidate for integration between systems and applications from BPs that are candidate for BI, enabling the BPs that candidate for BI to be optimized and analyzed based on the modelled and integrated BPs.
- Service-oriented Business Process Communication Layer (SoBPCL) is proposed as a universal way of communication that enables the connection and interaction among existing software systems based on SOA. Therefore, TWO prototypes were developed and deployed in SoBPCL: 1) the SoBPFA, which integrates the BPs of internal and external software systems. It leverages existing BP and BI functionalities and user interfaces while avoiding extensive redesigns of existing systems and architectures. 2) The SoICA integrates the internal BPs among existing systems, coordinates the distributed transactions, and maintains the data interoperability when the transactions and queries span multiple SOA services. For example, SoICA integrates the internal BPs that flow between heterogeneous software systems. In addition, the SoICA supports adaptability by loosely coupling among the existing systems. If an organization decides to upgrade BPM tool or if the BPM tool failed, existing systems must remain operational, and their BPs are internally integrated using SoICA module and without a BPM.

In terms of BPI, the proposed SO2PI model expands the SOA integration layer to include the following two types of integration: 1) SoICA integrates the internal BPs among existing systems, coordinates the distributed transactions, and maintains the data interoperability when the transactions and queries span multiple SOA services. For example, SoICA integrates the internal BPs that flow between heterogeneous software systems. The integration services of SoICA are exposed into ESB that can be accessed locally from heterogeneous systems. The SoICA is developed and implemented using direct access services, REST API, and Java Script Open Notation (JSON); 2) SoBPFA integrates the external BPs, for example, the BPs that flow between internal and external system. In addition, the SoBPCL layer can be described as a custom extension to the ESB integration, specially adapted to the requirements of the proposed SO2PI model. It is a middleware layer designed to support the integration and control of BPs that are managed by existing software systems. While typical SoBPFA transports and translates data between existing software systems through the tasks of integrated BPs, the SoBPCL also has its own data domain model to simplify business data. Principles like the auto generation of SoBPCL REST client artifacts enable the compatibility of a SoBPCL with existing software systems developed in different languages, tools, and technologies. The compatibility of the SoBPFA implementation with the existing systems leads to a low-effort integration into the existing architecture (Zhang & Yue, 2020). As for BP modeling, optimization, and execution, the proposed model builds an Educational Business Process Flow Application (SoBPFA), which connects and interacts with existing software systems based on SOA. The major functions of So2PI model are shown in Fig. 1 below.

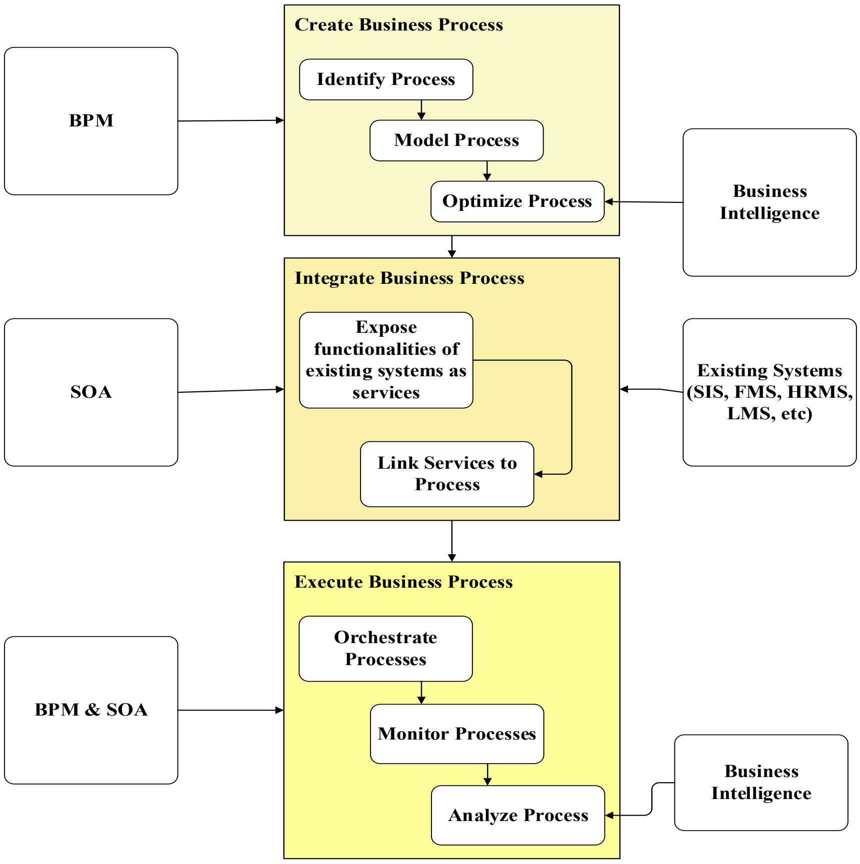

Fig. 1: So2PI major functions

3.3. The Roles of Users and Developers in the SO2PI Model

There are three roles: business analyst, integration developer, and business user. The business analysts log into SoBPFA to identify the BPs that are candidates for integration and then design their workflow. Integration developers design and expose the SOA services based on the functionalities of the existing systems. They also work together with the business analyst to link the SOA services to the activities of the BPs that are designed in SoBPFA. Business users use the existing systems to execute the integrated BPs, particularly those that require manual interaction (i.e., approve, reject, and so on).

3.4. So2PI Layers and Components

The So2PI consists of flowing layers and components:

1) **So2PI Service Layer**: this layer includes several types of services as follows:

- Integration Services: The Integration Services are responsible for the internal and external integration among the various types of existing software systems (i.e., SIS, FMS, LMS, and HRMS). It consists of the following parts:

i. Integration REST Services: The Representational State Transfer (REST) allows communication between two software systems over the Internet using REST APIs.
ii. Direct Access to Databases: This type of integration service provides direct access to databases for internal integration among various types of existing software systems. For example, integrating a payroll module with an accounting system may require the use of a set of pre-integration services to manage and control the payroll process. This fundamental type of integration is designed to ensure the underlying databases' speed and availability.

- External Integration Service: This integration service integrates, for example, between the in-house SIS and the third party such as PayPal, for enabling the students to pay the tuition fees online.
- Business Intelligence Services: This type of service is used to monitor the execution on integration services and provides statistical information about running, stopped, process state updates, notifications of various events. This statistical information is shown to the end users via dashboard.
- General Services: General Services (GS) are responsible for monitoring and authorizing access to various types of existing software systems (i.e. SIS, FMS, LMS, and HRMS). It includes Authorization and Security Services. A common example is the use of a Validation Authority (VA) server to verify user certificates while performing key actions in the applications. These actions include authorization during initial login, order execution, user authorization changes, and so on.

2) Service-oriented Integration and Coordination Adapter (SoICA): The Service-oriented Integration and Coordination Application (SoICA) integrates seamlessly with BPM based SOA to facilitate REST APIs integration. The SoICA enables connectivity between existing systems and SOA Services through lightweight integration. It provides the technology to automate business processes not only within the organization but also with customers, partners, and suppliers. Furthermore, SoICA addresses integration issues by leveraging process-to-process connections, as well as creating internal and external API connectivity with existing systems and SOA Services without the need for complex code. SoICA is made up of two parts: an integration adapter and a transaction coordinator. It consumes adapter services exposed as Integration and BI Services from existing applications. The SoICA is an important part of SO2PI integration model that developed into existing systems, enabling their communication with SoBPFA. The basic tasks of the adapter are performed by the following four components:

- Client BP Executor (related their actions) - has the role of sending messages to SoBPFA and processing responses from SoBPFA, all with the help of SoBPCL, which is described in the next section. Communication according to SoBPFA is realized at certain points of the business process /and includes informing SoBPFA about changes in the state of BP. Communication with SoBPFA can include process instance creation, process state updates, notifications of various events, retrieving data from decision engines or decision tables, etc.
- Integration Functionality Exposure Component – exposes functionalities of existing systems as SOA Services to be used by component of Client Integration Services in SoBPFA. If necessary, it contains additional business logic that groups, manipulates and adjusts existing functionalities.
- BI Functionality Exposure Component – exposes functionalities of existing systems as SOA Services to be used by component of Client BI Services in SoBPFA.
- Authorization and Redirection Component (ARC) – authorizes the user coming from SoBPFA to the existing software systems and can perform redirections on the existing software systems based on the information it receives from SoBPFA.

3) Service-oriented Business Process Communication Layer: Service-oriented Business Process Communication Layer (SoBPCL), as its name suggests, contains methods for communication between SoBPFA and existing systems applications. SoBPCL has the role of redirecting calls, already contains business logic, which enables it to perform security checks, data mapping, construction calls under SoBPFA, etc. SoBPCL contains information in which applications requested data are located, which SOA services should be called, and how to adjust the retrieved data for SoBPFA. There are several basic components defined in this layer:

- Services for managing process actions – They represent a component of SoBPCL through which the communication of existing systems according to SoBPFA takes place. Communication takes place regardless of the flow of BP and consists of specially defined services specialized in performing general actions. An action which can be starting a process instance, changing the status and data in the process, ending a specific process activity, etc. contains:
  i. Client for process-related communication - a component for manipulating processes, activities and process data on SoBPFA. It calls the process APIs that are exposed in the SoBPFA environment.
  ii. Business logic related to process actions - serves for grouping and chaining process API calls on SoBPFA, and interpreting data from legacy systems.
  iii. Services for managing process actions are called by the SoBPFA client component, which is part of the adapter built into existing software systems.

- Services for working with existing functionalities – A component through which SoBPFA communicates with existing systems and, if necessary, checks or retrieves data. It contains:
  i. Client for communication with existing systems – consists of methods for retrieving data from existing systems, i.e., service calls that are exposed by the SoIA (a component for exposing existing functionalities).
  ii. Business logic for interpreting existing application data – serves to interpret, group and map the data of existing system according to SoBPFA needs.

- Simplified domain model – contains simplified business objects of the existing systems domain that describe the business area in which the organization is engaged. It is used by the service for working with existing functionalities and the service for managing process actions. This model, significantly simplified, is created by mapping the domain model of client applications so that it contains only the basic set of information needed by the BPM solution. In addition, there are also entities here that represent artifacts from the BPM environment, and are used when communicating with the BPM solution, for example entities of processes, activities, users, etc.

4) Service-oriented Business Process Flow Application*:* Service-oriented Business Process Flow Application (SoBPFA) is built in a BPM environment oracle Apex tool, and it contains a model of the BPs that are implemented in the existing software systems (i.e. SIS, FMS, LMS, and HRMS). The flows of BPs and their subprocesses are created in SoBPFA, methods for communication with SoBPCL are created, user interfaces are constructed, and the necessary business logic is added. Business process management (starting processes and activities, updating data, assigning tasks, etc.) takes place through interfaces for process-related communication (an API component for process-related communication) that are exposed in the BPM environment. These interfaces are part of the BPMS used by the organization and depend on the manufacturer's implementation. In general, they enable access to artifacts, user tasks, business processes, and BPs data in a BPM environment. In addition, SoBPFA contains a component for integration with existing systems that calls services over the integration adapter of existing systems via SoBPCL, which retrieves business-relevant data needed in certain steps of BPs. In this way, direct communication is achieved with the SOA services, and with the help of business logic on SoBPCL. Manipulation of the required business data can be done if necessary. At the level of the BPM environment, user authorizations and groups are defined, which are used within SoBPFA when defining parts of the BP diagram. Through its interfaces, SoBPFA provides a central point of user interaction and then redirects users to legacy applications that perform BP activities. Process data displayed on SoBPFA interfaces is read-only, while updating and changing process data is done exclusively through legacy application interfaces. The goal of the SO2PI model is to manipulate BP's data through the existing interfaces of existing systems and applications, which is expressed through the third characteristic of the SO2PI model. For the stated reason, complex data types used in existing applications are further simplified in SoBPFA, and mapped into simple types that are necessary for the functioning of SoBPFA. Consequently, the amount of data displayed on SoBPFA interfaces is greatly simplified with respect to the volume of data displayed in existing applications. It contains a basic set of information about the process, and it is defined in cooperation with business users who select the most business-relevant data about the process.

5) Business Process Identification Layer: The Business Process Identification Layer (BPIL) is used to identify and select the BPs that span across multiple software systems and applications. In the BPIL, the business rules are an important part that is used to identify and select the BPs that are candidates for integration processes. After the identification of BPs, their flows should be defined and modeled in the SoBPFA. As shown in Fig. 2, the BPs are categorized into three types, including integration, decision, and analysis processes.

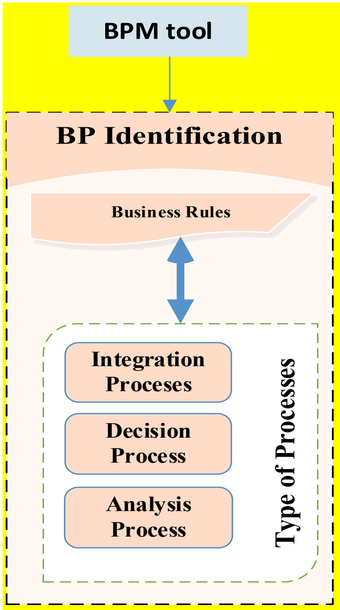

Fig. 2 BP Identification layer of SO2PI

The process category, subcategory, and examples of each category are described in Table 3 below.

Table 3. Categories of business process in the proposed SO2PI MODEL

Process Category	Process subcategory	Example
Decision related process	Simple decision-making process. Intelligent decision-making process	Pass to higher level manager. If within variance, then approve.
Analysis related process	BP analysis	View dashboard
Service-related process	Data service-related process Web service-related process Business service-related process	Access databases and transactions of existing systems

6) Data Model of the SO2PI

Table 4 below shows the data model of the proposed SO2PI Model and Fig 3. illustrates the proposed model.

Table 4. The database tables of the SOBPFA

Table Name	Table Description
SOBPFA_PROC_DOMAINS	Used to store the type of business domain. It helps in grouping the business process for each domain.
SOBPFA_PROC_DIAGRAMS	Used to store definition and metadata of the BP diagrams as xml files.
SOBPFA_PROC_CATEGORIES	Used to store the categories of business process such as decision related process, analysis related process, service related process
SOBPFA_PROC_TASKS	Used to define and store the tasks related to BPs and their flows
SOBPFA_PROC_OBJECTS	Used to define and store the metadata of objects of BP diagrams
SOBPFA_PROC_CONN	Used to define and store the metadata of connections of BP diagrams
SOBPFA_PROC_OBJECT_EXP	Used to define and store the metadata of objects expressions
SOBPFA_PROC_ACT_VARS	Used to define and store the variables of BP’s tasks
SOBPFA_PROC_SUBFLOWS	Used to define and store the sub flow of BP
SOBPFA_API_SCHEMA	Used to store the metadata for the existing applications and systems
SOBPFA_API_CATEGORIES	Used to store the metadata of BP’s categories
SOBPFA_API_SERVICES	Used to store the metadata and definitions of API services
SOBPFA_API_TASK_SERVICES	Used to store the metadata and definitions of API services linked to each task of BP
SOBPFA_PROC_OBJECT_ATTR	Used to define and store the attributes of objects

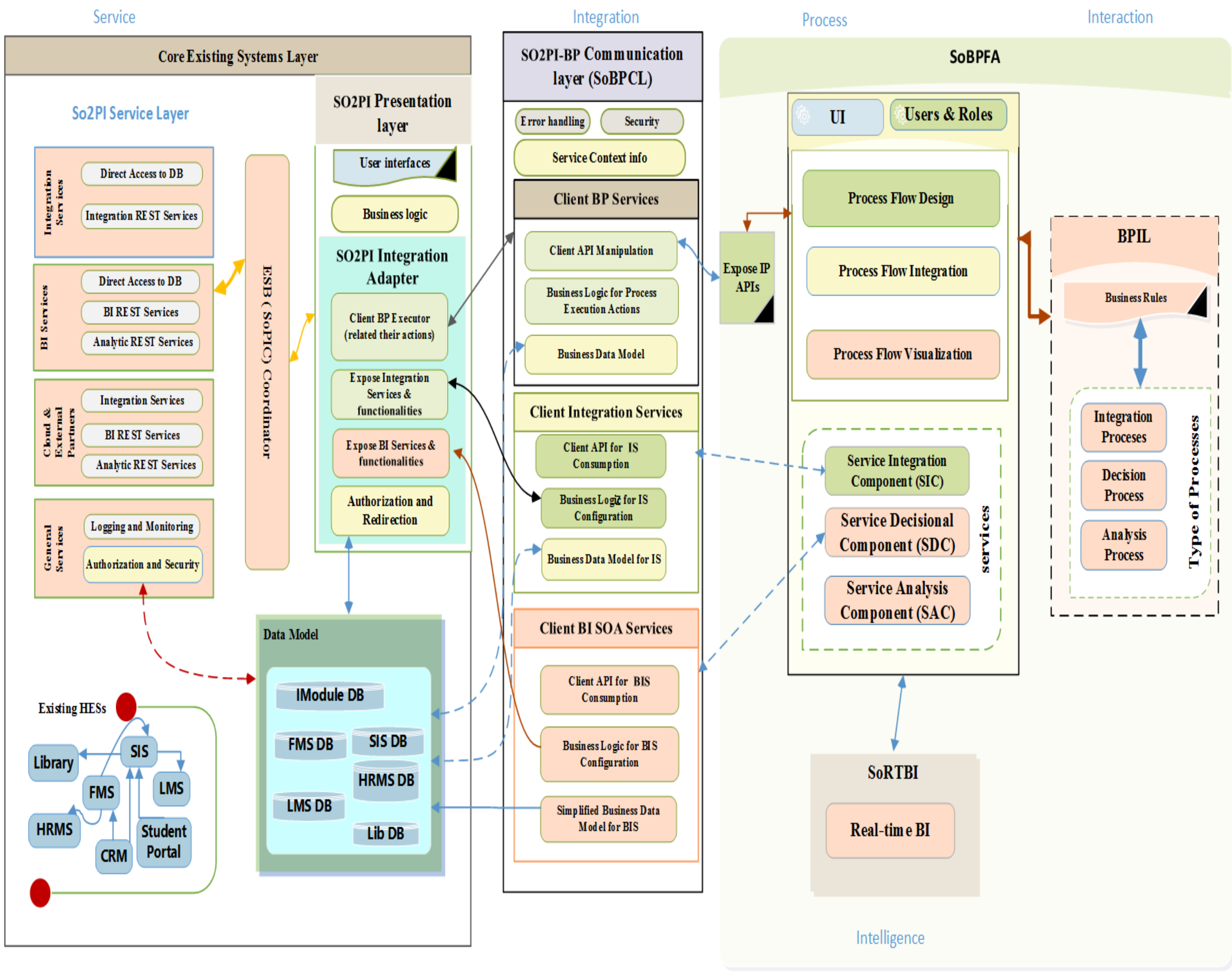

Fig. 3: The Proposed SOPI Model

4. Evaluation

The SO2PI model evaluation includes two parts. First, the developed prototypes were validated, and their capabilities were demonstrated using a case study (Sect. 5.1). Second, assessment of the SO2PI model (Sect. 5.2) by providing explanation for the selected evaluation procedures (expert evaluation), followed by their application.

4.1. Validation through Case Study

Due to the widening range of functionality, the integration module for integrating the BPs is becoming more complex, and there are several problems related to integration requirements from IR1 to IR7. Therefore, a review of the existing systems and their architectures in the environments of the case study is critical for identifying the BPs that need to be integrated across several heterogeneous systems. At this point, the proposed SO2PI is implemented in a 'real-world' environment using a case study, where it interacts with actual process-oriented systems to systems and systems to humans that initiate and manage BPs. The validation of the proposed SO2PI model is based on SoBPFA and SoICA modules developed in Sections 4.2 and 4.4. This validation is to ensure that the development of the BP integration modules meets the integration requirements from IR1 to IR7. In line with the desire of the case study environment to integrate the BPs based on the functionalities of existing software systems without making significant redesigns, the IT manager selected five (5) of the most experienced IT practitioners from the IT Center of the University of Science and Technology (UST) to join the researcher in the verification of the proposed SO2PI. The verification process was a six-month exercise at the university. Table 5 shows the selected criteria for implementing the case study in the university.

Table 5. SO2PI environment selection criteria result

Criteria description	Criteria
Implements the heterogeneous systems and applications	Student Information System (SIS) Human Resource Management system (HRMS) Learning Management System (LMS Moodle) Finance Management System (FMS) Library system
Number of students	> 100000
Number of employees	> 1000

The next subsection describes the validation for the integration requirements:

1) Validating the Integration Requirements (IR3, IR6, and IR4): A screenshot of a Course Creation and enrollment BP (CEBP) that was created by SoBPFA as shown in Fig. 4. The CEBP is used to validate the IR3, IR6, and IR4. As a result, the integration of CEBP is implemented using the SoBPFA. The course is approved in the SIS system by the user who has the responsibility to approve the task. Based on the data available in the CCEBP (status and type of the CCEBP), the SoBPFA asks the SoBPCL to determine and create a link to the Process Integration Adapter for starting the integration process based on the rules and defined flow. Therefore, the integration adapter responses to SoBPFA by making link to LMS system using makeLinkToTaskInExistingSystem API.

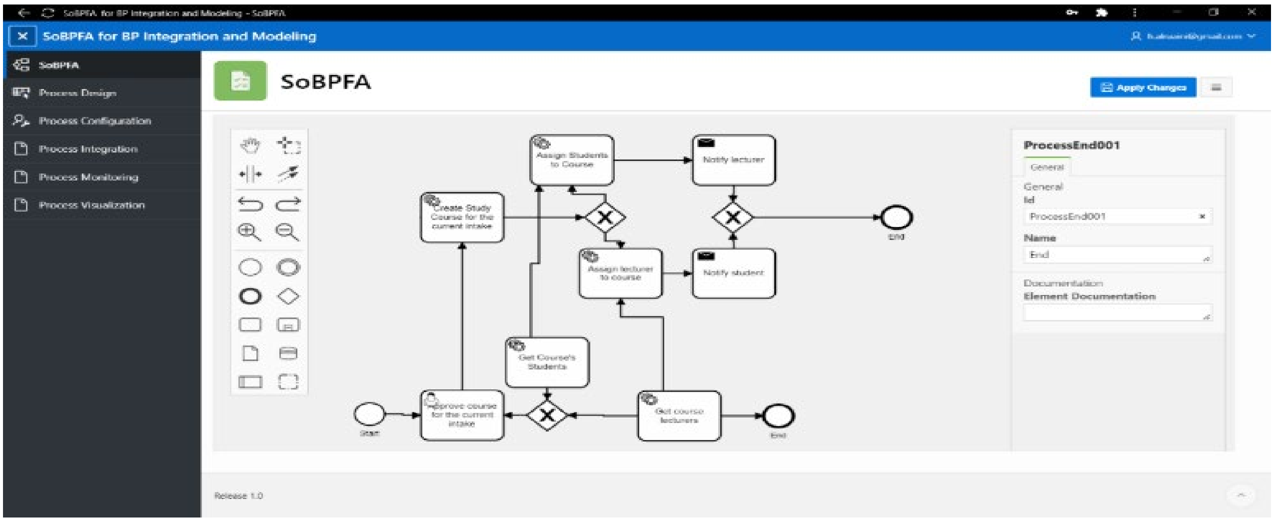

Fig.4: The BP model for integrating the SIS and LMS systems.

2) Validating the Integration Requirements (IR1 and IR4): The main Restful APIs mentioned in Table 4 are used to validate IR1, IR4. In addition, Restful APIs are used to execute and control the BP integration by execution the specific API mentioned in Table 6.

Table 6. The main rest API of SOBPFA are used to validate IR1, IR4

Rest API Name	Usage
findAndStartProcessInstance	Used to create process instance and start process execution
findAndAssignTaskToUorA	Used to start the first task of the process
makeLinkToTaskInExistingSystem	Make link to specific task in existing system
findAndFinishTaskFromUorA	Used to finish the task of the process
updateProcessInstanceData	Used to updates the status of the process

The SoBPFA enables the authorized user to perform the manual steps and tasks of the CCEBP. The authorized user login to SoBPFA to perform approval task for the study course in specific intake, then the SoBPFA will communicate with integration adapter to retrieve the data from SIS system and transfer these data to the LMS system as shown in Fig. 5 and Fig. 6.

Table 7. Business tasks and objects on the business level are used to validate IR1, IR4

Existing Systems	Identified business tasks	Business Objects	SOA Services
SIS	Start creating a study course in a specific semester for the specified study program. Get semester info for the course. Get branch info for the course. Create a study course for the specified semester. Assign lecturer to the course. Assign students to the course	*UniqueProcessId* *CourseId* *SemesterId* *LecturerId* *StudentId*	*GetSemesterService* *GetStudyProgramService* *GetBranchService* *AssignLectureService* *AssignStudentService* *GetLectureService* *GetStudentService*
HRMS	Get lecturer information		
LMS	Create course enrollment. Open new accounts for lecturer and students Assign lecturer to course. Assign students to the course		

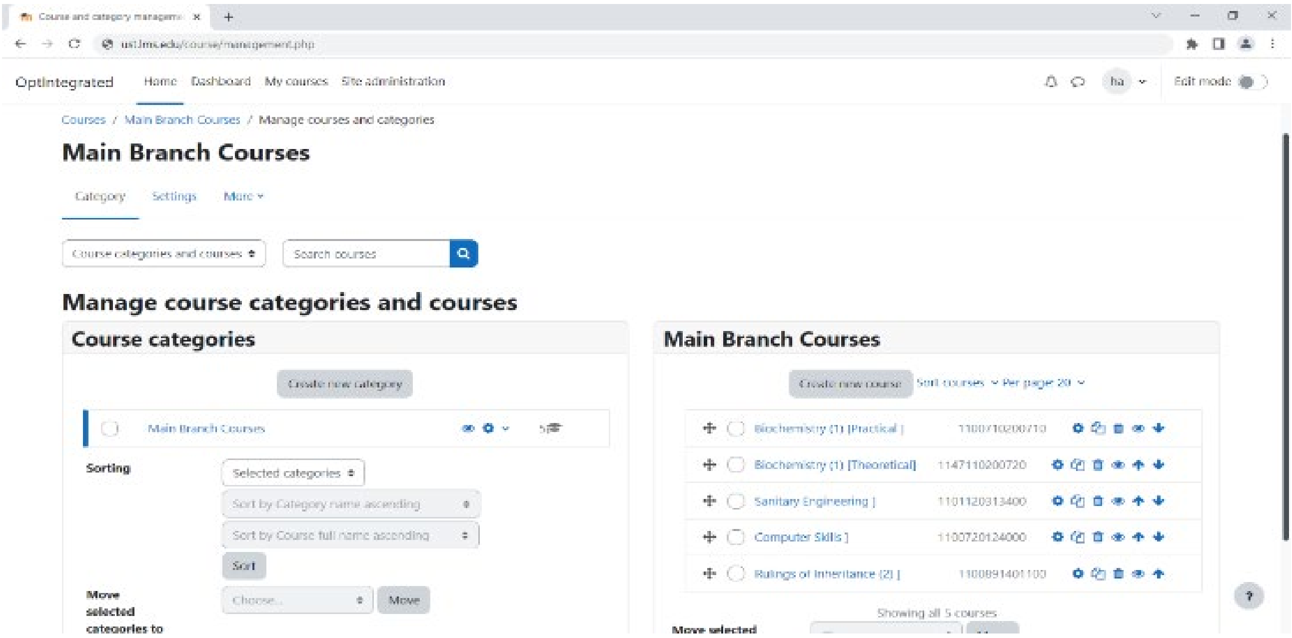

Fig. 5: Results of BPI –the courses posted from SIS to LMS

This case study demonstrates that the SoBPFA prototype indeed integrates among three heterogeneous systems: SIS, HRMS, and LMS. The steps of integration in this use case are as follows: 1) The business analysts login to the SoBPFA and then design the BPs that need to be integrated; 2) the integration specialists apply the SOA services to the tasks of the integrated BP as mentioned in Table 2; and 3) the SoBPFA directly utilizes the BP model to execute the integrated BP (called CCEBP) and its instances to submit data between the targeted systems. SoBPFA also shows that changes to the BP design are automatically reflected in the instance model.

3) Validating the Monitoring and Real-time Capability (IR5)*:* As shown in Fig. 6 and Fig. 7, the execution of integrated BPs of existing software systems and applications are monitored in real time. Accordingly, not only the integration requirement IR5 is validated but also IR6 and IR3.

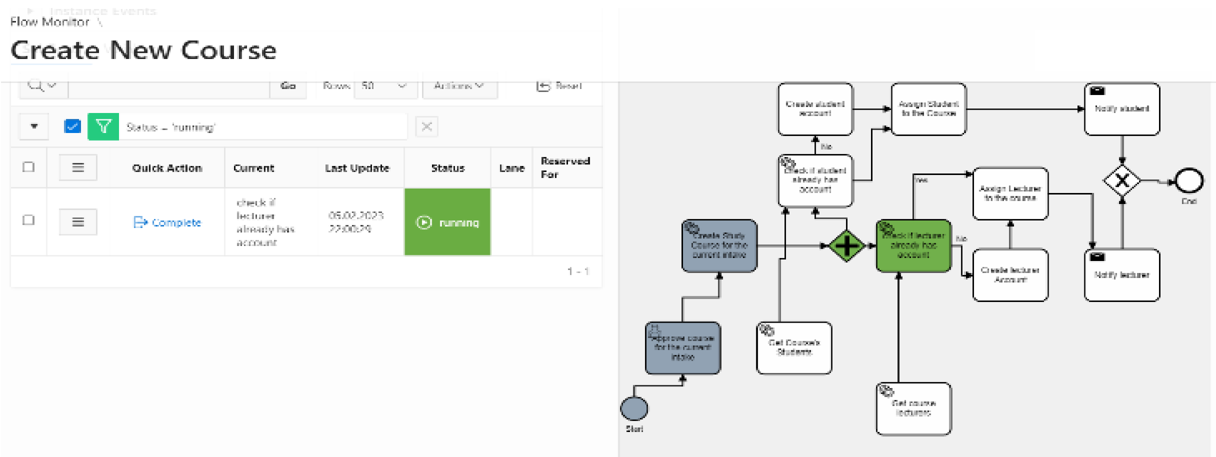

Fig. 6: monitoring the execution of business processes.

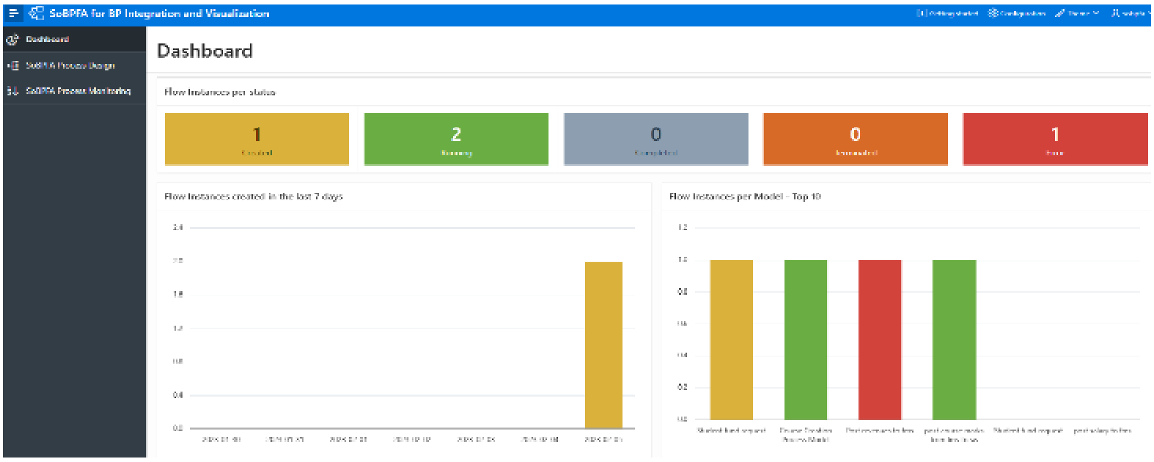

Fig. 7: Real-time Business Intelligence

4) Validating the Data Interoperability (IR1)*:* The validation of data interoperability ensures that all transactions related to the activities of BP are committed or rolled back, as shown in Fig. 8 below.

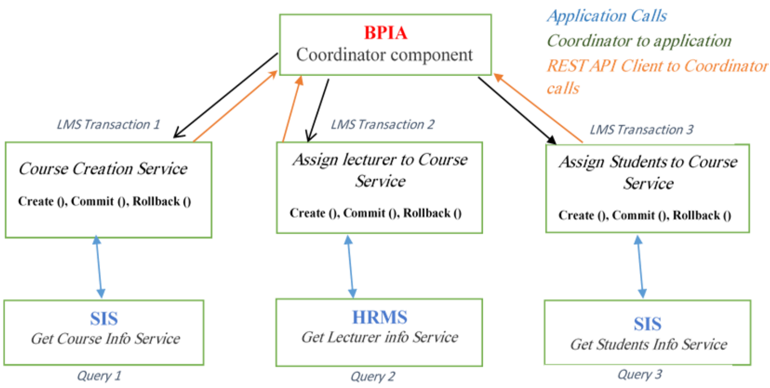

Fig. 8: Data interoperability between the SOA Services linked to BPs (IR1)

4.2. Assessment of SO2PI

According to Frank (2006) and Schelp and Winter (2006), research that aimed at developing models as artifacts must be scientifically justified. The required justification is difficult because the successful application of such a method of integration and modeling is dependent on a variety of factors related to not only the experience and qualifications of users but also to their attitudes and learning curve. Furthermore, where such an artifact is novel, practitioners and users may lack a clear understanding of current and future implementations of a BP integration model, making it difficult to assess its practical benefits. To address these concerns, this study is based on a justification approach, which proposes two major research process guidelines: multi-criteria justification and transparency of assumptions such as discursive evaluation (Frank 2006). Transparency needs identifying all non-trivial assumptions that are not clear or can be inferred from established knowledge, as well as evaluating the model against the integration requirements. In addition to consensus, discursive evaluation meets several prerequisites, including openness, traceability, knowledge and experience, the absence of hidden political preferences, and the absence of coercion or incentives that influence the preferences of certain arguments (Habermas 1984). Although reaching a consensus in a reasonable discussion does not ensure the absolute truth, it may be the only way to develop an evaluation or justification when correspondence or coherence theory appear inapplicable or insufficient. Ideally, multiple dedicated experts perform rational discourse; however, a virtual discourse, a relaxed form of discursive evaluation based on carefully collecting and encountering arguments from various perspectives, is also possible. As previously stated, the current research's justification procedures are divided into two parts: The first, requirement-based evaluation is a hybrid of virtual discursive evaluation and the coherence theory of truth (i.e., the assumptions can be substantiated by reference to a body of existing literature). The second part is an evaluation of rational discourse by experts and IT practitioners. The following subsections describe these justification procedures.

4.3. Expert Evaluation

The expert evaluation has focused on three major objectives: (1) evaluating the relevance of the dynamic integration requirements; (2) evaluating the SO2PI model; and (3) obtaining feedback on further development and customization of the SO2PI model. Ten experts participated in the evaluation of the proposed SO2PI. The selected experts work as SOA architects and IT consultants in five different software companies and have at least ten years of experience in software systems integration, process identification and design, service identification, development of process-oriented integration modules, or development of BPM applications and tools. The procedure of evaluating the SO2PI included the following steps. First, the aims of the research as well as the dynamic integration requirements from IR1 to IR7 were presented. Then, the developed prototypes were demonstrated to experts. A 4-hour face-to-face presentation was conducted before receiving the feedback from the expert. The feedback was guided by a set of thirty closed questions. These questions were carried out as an open discussion to enable the participant(s) to add ideas and comments beyond the guiding questions. Responses were indicated on a Likert scale (Strongly Disagree (SD), Disagree (D), Undecided (U), Agree (A), and Strongly Agree (SA) were coded with an ordinal value of 1, 2, 3, 4, and 5). The responses and comments that were obtained from the participants regarding the questions were compiled and summarized. Fig. 8 below shows the results of experts' evaluation.

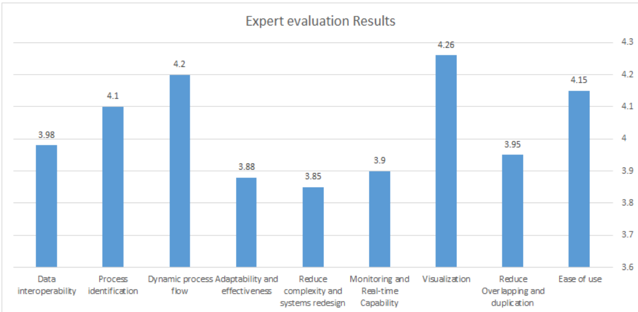

Fig. 9: Expert evaluation results

With respect to the data interoperability between the activities of BPs, Fig. 9 shows a decision mean of 3.98, which indicates that the general opinions of the experts agree with the ability of the proposed SO2PI model to integrate between the functionalities of existing software systems that qualify as candidate integration services and the activities of the BPs at runtime. In addition, with respect to the identification of BPs, Fig. 9 shows a decision mean of 4.10, which indicates that the general opinions of the experts agree with the ability of the proposed SO2PI model to identify the functionalities of existing software systems that qualify as candidate integration services and BI services. Similarly, a decision mean of 4.10 indicates the ability of the proposed SO2PI model to identify the BPs that should be integrated at runtime. Moreover, with respect to effectiveness and adaptability, Fig. 9 shows a decision mean of 3.88, which indicates that the general opinions of experts agree that the proposed SO2PI has achieved the adaptability to fit into the evolving and changing information requirements by using an approach of modular functionality of the existing systems.

Additionally, with respect to dynamic process flow, Fig. 9 shows a decision mean of 4.20, which indicates that the general opinion of experts is that the proposed SO2PI has achieved the scalability requirement. Furthermore, with respect to the real-time capability-related BP workflow, Fig. 8 shows a decision mean of 3.90, which indicates that the general opinion of experts is that the proposed SO2PI model has reduced decision latency by monitoring the workflow of BPs, analyzing the complexity of BPs, and proving real-time information for optimizing the efficiency of integration and decisional processes. Therefore, it is necessary for the decision makers to quickly decide about the BPs that require improvement based on the market's demands.

With respect to reducing complexity and the significant redesign in existing systems, the Fig. 9 shows a decision mean of 3.85, which indicates that the general opinions of experienced IT practitioners and users agree that the proposed SO2PI has achieved good reduction in integrating the complex BP as well as minimizing the systems redesign and maximizing resource utilization. These two major requirements are very important to tackling the problems mentioned in the Problem Statement Section. Therefore, complexity and reduction are required to provide the dynamic BP integration, while the reduction of existing systems' redesign is required to reduce the cost and effort of implementing BP integration.

Also, with respect to the reduction of overlapping and duplication, the result shows a decision mean of 3.95, which indicates that the general opinion of experts agrees that the proposed SO2PI model has reduced the overlapping and duplication of the similar SOA service and integrated process. Finally, with respect to the visualization of the BPs, the result shows a decision mean of 4.26, which indicates that the general opinion of experts agrees that the execution of the BPs is visible to end user.

The effectiveness would help the SO2PI model to achieve the interoperability between the disconnected BPs, which are performed by different heterogeneous systems without significant redesign in existing systems, and applications. In addition to utilize the functionalities of existing systems and integrating business processes in heterogeneous environment. Furthermore, the effectiveness of SO2PI participated in directly utilizing the BP model to effectively control and manage execution of BPs together with their integration flows across multiple of existing systems and serving as an efficient way to visualize execution processes in existing systems.

5. Conclusion

The proposed SO2PI (Service-Oriented Dynamic Business Process Integration) model offers a comprehensive solution for integrating internal and external Business Processes (BPs) with distinct advantages over traditional approaches. The model encompasses seven integration requirements that align with its five key components: interaction, service, process, integration, and intelligence. These requirements were derived from existing literature and feedback from software industry consultants, professionals, and researchers, ensuring consensus on the dynamic BPI requirements.

The SO2PI model's validation was conducted through the development of prototypes: SoBPFA and SoIPC, and their capabilities were successfully demonstrated using a practical case study. The SO2PI model stands out in several critical aspects: firstly, it supports direct utilization of the BP model for integrating business processes across multiple software systems, allowing dynamic modification of integrated BP flows at runtime without impacting implementation. Secondly, it facilitates the identification of reusable functionalities in existing systems based on BP tasks and maintains data interoperability through BP task data mapping between software systems. Thirdly, the model reduces redundancy and maximizes the reuse of services and processes, streamlining the integration process. Lastly, it enables real-time monitoring of integrated business process execution, offering performance analysis at runtime.

Despite the significant contributions of the SO2PI model, there are avenues for further research. To enhance real-time Business Intelligence (BI), future investigations should focus on incorporating real-time features related to existing systems and their architectures, not just workflow characteristics. Additionally, exploring additional visualization types, particularly abstract representations to support individual adaptations, warrants further exploration. The experts' feedback also indicates a need to automate new BPs that are not currently managed by existing systems, although this aspect is not a primary focus of this research.

In conclusion, the SO2PI model presents a promising advancement in dynamic BP integration, offering enhanced adaptability, efficiency, and real-time monitoring capabilities. Further research and refinement will be instrumental in expanding the model's capabilities and solidifying its position as an effective tool for seamless and agile business process integration across software systems.

References

Abai, N. H. Z., Yahaya, J. H., & Deraman, A. (2017). The determinants of integrated business intelligence and analytics in organizational performance process. In 2017 6th *International Conference on Electrical Engineering and Informatics (ICEEI) (pp. 1-6). IEEE*.

Abdellatif, M., Shatnawi, A., Mili, H., Moha, N., El Boussaidi, G., Hecht, G., & Guéhéneuc, Y. G. (2021). A taxonomy of service identification approaches for legacy software systems modernization. *Journal of Systems and Software, 173, 110868*.

Abdullah, H., Taa, A., & Mohammed, F. (2021). Multi-domain Business Intelligence Model for Educational Enterprise Resource Planning Systems. *In International Conference of Reliable Information and Communication Technology (pp. 1218-1224)*. Springer, Cham.

AbouEl-Seoud, S., AboGamie, E. A., & Salama, M. (2017). Integrated education management system via cloud computing. *International Journal of Interactive Mobile Technologies* (iJIM), 11(2), 24-33.

Al-Barakati, A. (2021). Leveraging Artificial Intelligence–enabled Workflow Framework for Legacy Transformation. *International Journal of Advanced Computer Science and Applications*, 12(12).

Aldossari, S., & Mokhtar, U. A. (2020). A model to adopt enterprise resource planning (ERP) and business intelligence (BI) among Saudi SMEs. *International Journal of Innovation*, 8(2), 305-347.

Ali, A. A., & Mohamed, W. M. (2018). Using a Business Activity Monitoring and SOA for a Real-Time ETL. Int. J. Comput. Appl, 180(10), 975-8887.

Belchior, R., Guerreiro, S., Vasconcelos, A., & Correia, M. (2020). A survey on business process view integration. arXiv preprint arXiv:2011.14465.

Belete, G. F., Voinov, A., & Laniak, G. F. (2017). An overview of the model integration process: From pre-integration assessment to testing. Environmental modelling & software, 87, 49-63.

Borse, A. A., Verma, S., Babu, S., & Kumar, G. N. (2019). Service oriented architecture paradigm for business intelligence: A survey. *International Journal of Advance Research, Ideas and Innovations in Technology*, 5(1), 196-199.

Brian, T., Muramuzi, D. & Kanyunyuzi, C. (2019). Integration of Service Oriented Architecture into Enterprise Applications: A Process Approach.

Butt, J. (2020). A conceptual framework to support digital transformation in manufacturing using an integrated business process management approach. Designs, 4(3), 17.

Chalmeta, R., & Pazos, V. (2015). A step-by-step methodology for enterprise interoperability projects. Enterprise Information Systems, 9(4), 436-464.

Correia, A. M. M., Rocha, C. F., Duclós, L. C., & Veiga, C. P. D. (2021). Integration of business Processes with activities and information: Evidence from Brazil. SAGE Open, 11(1), 21582440211006135.

Daraghmi, E., Zhang, C. P., & Yuan, S. M. (2022). Enhancing Saga Pattern for Distributed Transactions within a Microservices Architecture. Applied Sciences, 12(12), 6242.

Dell Boomi (2018). Dell Boomi for higher education, Build your connected digital campus with cloud integration, retrieved from https://boomi .com/wp-content/uploads/Solution-Brief-Higher-Ed.pdf

Estefania, T. V., Samir, L., Robert, P., Patrice, D., & Alexandre, M. (2018). The integration of ERP and inter-intra organizational information systems: A Literature Review. IFAC-PapersOnLine, 51(11), 1212-1217.

Fajar, A. N., & Legowo, N. (2018). Services modeling based on SOA and BPM for information system flexibility improvement. *International Journal of Electrical and Computer Engineering*, 8(4), 2451.

Fajar, A. N., Nurcahyo, A. & Sriartnasari, S. R.(2018). SOA system architecture for interconnected modern higher education in Indonesia,3rd *International Conference on Computer Science and Computational Intelligence 2018*, Procedia Computer Science 135 (2018) 354–360.

Frank U (2006) Towards a pluralistic conception of research methods in information systems research

Goldstein, A., Johanndeiter, T., & Frank, U. (2019). Business process runtime models: towards bridging the gap between design, enactment, and evaluation of business processes. Information Systems and e-Business Management, 17(1), 27-64.

Gonzalez-Lopez, F., & Bustos, G. (2019). Business process architecture design methodologies–a literature review. *Business Process Management Journal*, 25(6), 1317-1334.

Habermas J (1984) The theory of communicative action, vol 1. Beacon Press, Boston.

Hasic, F., Vanwijck, L., & Vanthienen, J. (2017, November). Integrating Processes, Cases, and Decisions for Knowledge-Intensive Process Modelling. In PrOse@ PoEM.

Hrabala, M, Opletalova, M. & Tomas,B. (2017): Business Process Management In CZECH Higher Education

https://www.mulesoft.com/resources/cloudhub/ipaas-integration-platform-as-a-service.

Jakimoski, K. (2016). Challenges of interoperability and integration in education information systems. *International Journal of Database and Theory and Application*, 9(2), 33-46.

Javidroozi, V., Shah, H., & Feldman, G. (2019). A framework for addressing the challenges of business process change during enterprise systems integration. Business Process Management Journal.

Kaburuan, E. R., JM, H.O, Suprapto, F. & Januar, D. (2019): The combination of cooperation between SOA and BPM for School Management System, International Journal of Scientific & Technology Research Volume 8, Issue 10, October 2019

Kähkönen, T., (2017). Understanding and Managing Entreprise Systems Integration. Ph.D. Thesis, Faculty of Science and Technology, University of Technology, Lappeenranta, Finland on the 12th of May, 2017.

Kaula, R. (2020). Operational Intelligence Performance Inferencing across Business Processes. *International Journal of Database Management Systems* (IJDMS) Vol.12, No.1, February 2020

Kintz M., Kochanowski M. and Koetter F. (2017). Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach. DOI: 10.5220/0006135203530361 In Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017), pages 353-361

Kopp, A., & Orlovskyi, D. (2019, June). Intelligent Support of the Business Process Model Analysis and Improvement Method. *In International Conference on Information and Communication Technologies in Education*, Research, and Industrial Applications (pp. 111-135). Springer, Cham.

Linthicum, D. S. (2017). Cloud computing changes data integration forever: What's needed right now. IEEE Cloud Computing, 4(3), 50-53.

Ma, C., & Molnár, B. (2019). A Legacy ERP System Integration Framework based on Ontology Learning. In ICEIS (1) (pp. 231-237).

Matejaš, M., & Fertalj, K. (2019). Building a BPM application in an SOA-based legacy environment. Computer Science and Information Systems, 16(1), 45-74.

Mircea, M. (2012). SOA adoption in higher education: a practical guide to service-oriented virtual learning environment. *Procedia-Social and Behavioral Sciences*, 31, 218-223.

Ovum (2017). Getting Engaged: Institutions Shifting to Learner-Centric Administrative Systems: Fully integrated engagement is critical to building lifelong learning relationships

Paniagua, C., Eliasson, J., & Delsing, J. (2019, February). Interoperability mismatch challenges in heterogeneous soa-based systems. *In 2019 IEEE International Conference on Industrial Technology* (ICIT) (pp. 788-793). IEEE.

Raminder, P. S. & Sangeeta, A. (2018). ERP challenges in higher education, International Journal of Management and Applied Science, ISSN: 2394-7926, Volume-4, Issue-2, Feb.-2018

Ray, P., & Kumar, T. A. (2019). Analyzing the interface of business intelligence and enterprise resource planning.

Russman, R., Seymour, L. F., & Belle, J. P. (2017). Integrating BI Information into ERP Processes-Describing Enablers. *In International Conference on Enterprise Information Systems* (Vol. 2, pp. 241-248). SCITEPRESS.

Santoso, L. W. (2018). Integration between ERP software and business intelligence in odoo ERP: Case study A distribution company (Doctoral dissertation, Petra Christian University).

Schelp J, Winter R (2006) Method engineering: lessons learned from reference modeling. In: Chatterjee S, Hevner A (eds) *Proceedings of the first international conference on design science research in information systems and technology* (DESRIST 2006), pp 555–575

Serrano, M. A. G., Pérez-Marín, D., & Alarcón, M. I. (2014). Architecture Proposal Model for SOA in Universities Educational Software. *Asian Journal of Technology & Management Research* [ISSN: 2249–0892], 4(02).

Shi, Z., & Wang, G. (2018). Integration of big-data ERP and business analytics (BA*). The Journal of High Technology Management Research*, 29(2), 141-150.

Somya, R., Manongga, D., & Pakereng, M. A. I. (2018). Service-Oriented Business Intelligence (SoBI) for Academic and Financial Data Integration in University. *In 2018 International Seminar on Application for Technology of Information and Communication* (pp. 1-5). IEEE.

Srimathi, H., & Krishnamoorthy, A. (2019). Integration of student system using iPaaS. *International Journal of Scientific and Technology Research*, 8(9), 602-606.

Talaoui, Y., & Kohtamäki, M. (2020). 35 years of research on business intelligence process: a synthesis of a fragmented literature. Management Research Review.

Trieu, V. H. (2017). Getting value from Business Intelligence systems: A review and research agenda. Decision Support Systems, 93, 111-124.

Vinaja, R. (2018). Enterprise resource planning and business intelligence systems for information quality: an empirical analysis in the Italian setting.

Wang, N., & Lee, V. (2011, April). An integrated BPM-SOA framework for agile enterprises. *In Asian Conference on Intelligent Information and Database Systems* (pp. 557-566). Springer, Berlin, Heidelberg.

Xu, W., Furie, D., Mahabhaleshwar, M., Suresh, B., & Chouhan, H. (2019). Applications of an interaction, process, integration and intelligence (IPII) design approach for ergonomics solutions. Ergonomics.

Zafary, F. (2020) Implementation of business intelligence considering the role of information systems integration and enterprise resource planning. *Journal of Intelligence Studies in Business*. 10 (1) 59-74

Zhang, X. & Yue, W. T. (2020). Integration of On-Premises and Cloud-Based Software: The Product Bundling Perspective, *Forthcoming in Journal of the Association for Information Systems*

Zhao, W. (2017, September). Analysis on the Integration of Business Intelligence and Business Process Management. *In 2nd International Conference on Judicial*, Administrative and Humanitarian Problems of State Structures and Economic Subjects (JAHP 2017) (pp. 346-349). Atlantis Press.