HAL
open science

Improving ERPs Integration in Organization: An EOS-Based GreneOS Implementation

Joseph Rahme, Bharat Masimukku, Nicolas Daclin, Grégory Zacharewicz

To cite this version:

Joseph Rahme, Bharat Masimukku, Nicolas Daclin, Grégory Zacharewicz. Improving ERPs Integration in Organization: An EOS-Based GreneOS Implementation. Computers, 2022, 11 (12), 10.3390/computers11120171 . hal-03879981

HAL Id: hal-03879981

https://hal.science/hal-03879981

Submitted on 30 Nov 2022

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

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

computers

MDPI

Article

Improving ERPs Integration in Organization: An EOS-Based GreneOS Implementation

Joseph Rahme [1], Bharat Masimukku [2], Nicolas Daclin [3] and Gregory Zacharewicz [3,*]

[1] Faculty of Arts and Sciences, American University of Beirut, Beirut 1107 2020, Lebanon
[2] Grene Robotics, Hyderabad 500033, Telangana, India
[3] Laboratoire des Sciences des Risques (LSR), Institut Mines-Telecom, IMT Mines Ales, CEDEX, 30319 Alès, France
* Correspondence: gregory.zacharewicz@mines-ales.fr

Abstract: Current ERPs are still limited by cost, customization, implementation time, and interoperability with other systems. Even if cloud-based ERPs attempt to overcome these limits, they do not completely answer all of them. Based on that postulate about recent ERPs, a conceptual architecture, technical architecture, and implementation architecture of an Enterprise Operating System (EOS) have been designed and proposed to address the services and functionality needed by Enterprise 4.0. This conceptual architecture describes the essential functions required in the EOS, while the technical architecture shows how these tasks cooperate to achieve the mission of the EOS. Among some implementation architectures proposed that benefited from the innovation and concept of the EOS, GreneOS has proposed a technical architecture motivated by EOS concepts. The purpose of this paper is to discuss the current interest, complementarity, and limitation of both the EOS conceptual architecture and its implementation into GreneOS to propose perspectives for the future developments of the EOS.

Keywords: enterprise operating systems (EOS); architecture; infrastructure; interoperability; enterprise operations; workflow; Archimate

Citation: Rahme, J.; Masimukku, B.; Daclin, N.; Zacharewicz, G. Improving ERPs Integration in Organization: An EOS-Based GreneOS Implementation. *Computers* **2022**, *11*, 171. https://doi.org/10.3390/computers11120171

Academic Editors: Pedro Pereira, Luis Gomes and João Goes

Received: 28 September 2022
Accepted: 17 November 2022
Published: 28 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction and Problem Statement

In previous works, the conceptual, technical, and implementation architectures of an Enterprise Operating System (EOS) have been designed to meet the services and functionalities recognized in the foundation paper [1,2]. The conceptual architecture described what core functions are required in the EOS, while the technical architecture showed how these implemented functions can cooperate to achieve the EOS mission. A first implementation architecture has been proposed to test the EOS innovation and concepts in a technical architecture based on Java HLA [3]. Meanwhile, GreneOS proposed another technical architecture inspired as well by the concepts of the EOS. The goal of this paper is to describe the process from concepts to implementation, providing the interest and current limits of the approach to propose perspectives for the further developments of the EOS.

For enterprises, ERP systems integrate organizations' applications at the informational and operational level to utilize and automate many back-office functions related to technology and information services by collecting, storing, managing, and interpreting business activity data. Over the years, these ERP systems offer increasingly improved enterprise application integration by providing the most integrated suite of applications to perform standard business functions [4]. At its most basic level, ERP software integrates these various functions into one complete system to streamline processes and information across the entire organization. The central feature of all ERP systems is a shared database that supports multiple functions used by different business units. Today, traditional on-premises ERP systems are gradually being eclipsed by the growing demand for cloud-based ERP systems [5] due to their lower costs and faster deployment. In this migration to cloud technology, the impact of the pervasive

digital transformation of businesses is a true challenge. The ERP module is basically presented as a black-box application implicating built-in services and modules. It does not ensure an efficient integration between its built-in applications [6].

Nevertheless, the current state of the art of ERP reports that interconnectivity (and interoperability) barriers remain between departments despite the use of advanced ERPs [7]. These interoperability barriers within an enterprise unit or department lead to limitations in communication with other departments. If a department encounters these limits, it can affect the overall efficiency of the enterprise. It is therefore important to choose an environment and a system that overcomes the interoperability barriers, and which will optimize the company as a whole and improve its efficiency.

Furthermore, the cost of an ERP remains high, and they are identified for some SMEs as too expensive of a system. Enormous upfront costs and per-user licensing is the biggest challenge with traditional enterprise resource planning systems. However, there are another unseen and scarier disadvantage in terms of:

- Data: the limited number of users on the platform results in incomplete, broken data and the loss of data precision.
- Culture adaptation: Every enterprise has its own culture, and this is most seen in its data, processes, and decisions, while the Degree of Customization is still extremely limited for strategic and technical orientation reasons.
- Integration: Enterprises today are bombarded with an array of applications for different processes, yet all of them at their core are incompatible with each other, resulting in data silos, lengthy workflows, and slow decisions based on legacy data. In addition, it is not possible to extract the business process model to go to simulation. The ERP module is presented as a black-box application implicating built-in services and modules which may not allow to master the efficient integration between its built-in applications and a direct intercommunication with enterprise resources.
- Decision-making process: As per a Gartner report, organizations lose a cumulative amount of 3% of their profits due to slow or delayed decision-making. Welcome to the new realm of enterprises, where real-time artificially intelligent decision-making is the norm, freeing humans from mundane repetitive tasks and leading to sustainable efficiency.
- Implementation/update: These platforms are becoming enormous in size and complexity; its implementation and updates are very time-consuming and involve excessive costs. As final reason, there is an extremely high implementation time.

So, enterprise information systems need more flexibility with explicit and agile workflows to react and quickly adapt to business changes. Then, there is a clear need for more standardization to better support interoperability. The user is still not an IT expert, so zero code is welcomed here. In addition, going to federated interoperability and frequently checking the advances and the support that can be brought by big data and AI is required as well. We must wonder what we can do now for the information system of the enterprise if it is enhanced by AI. As a dilemma, enterprises want to keep overcoming the ERP platforms, so white box and interoperability are frequently preferred versus AI features for the next generation of ERPs, and it is a current challenge and trade-off to fine-tune the embedding of AI in ERPs [8,9].

Consequently, today's ERP software domain remains looking for more flexibility and agility, making business workflows more efficient or able to react quickly and properly to business changes [10]. In addition, some recent results of the literature review article, e.g., proposed by [11] confirms that the quality of service still highly affects the information system interest and user satisfaction. In addition, the quality of information also affects the satisfaction of users of the information system. There is therefore a need for a platform based on the EOS, and the aim of this article is to present solutions that can achieve these objectives of flexibility and reactivity in the information systems of companies by providing:

- A unified platform;

- Democratization of data;
- Autonomous decision making;
- A bespoke operating system developed with Zero-Code.

To attain this, this paper develops a model-based approach from the conceptual level to the implementation. It will track how concepts are derived in the implementation of EOS components.

2. EOS Architecture Driven by Interoperability

To keep up-to-date, manufacturing enterprises need to use the latest results from the ICT sector, especially when collaborating with external partners in a supply chain and exchanging products and data. It must adapt quickly and deal with an increasing amount of heterogeneous information exchanged between partners, including machines (physical means), humans, and IT; in other words, it must develop interoperability.

As mentioned in the Introduction, interoperability—the ability of the interaction between enterprise systems—remains a barrier in enterprises that ERP intends to overcome. To consider all the dimensions of interoperability, numerous initiatives have been launched for the last two decades to go beyond its simple technical aspect. In that sense, the framework for enterprise interoperability [12] offers a structure guiding the development of concepts, methods, and tools for interoperability. Thus, the framework is composed in three dimensions, such as:

- Interoperability concerns as the levels in the enterprise where interoperability must be considered (business, process, service, and data);
- Interoperability approaches as the ways to develop interoperability and as defined in the ISO 14258 (integrated, unified, and federated);
- Interoperability barriers as the problem that an organization may face (conceptual, technological, and organizational).

As shown in Figure 1, the EOS embraces unified and federated methodologies from the incorporation to the full federation by encouraging 'on-the-fly' interoperations and supporting the correspondence and exchanges among heterogeneous and networked enterprises dependent on shared business references. It likewise endeavors to set up interoperability through the conceptual and organizational barriers in the four different levels of concerns by modifying business rules to control activities and tasks, encouraging the arrangement and real-time communication with enterprise operations, giving an instrument to control the authority of accessing the services, and providing a data distribution service for implementing on-the-fly mapping. Moreover, the interoperability approach is applied through the technological barrier and specifically on the level of data by ensuring database connection and remote access.

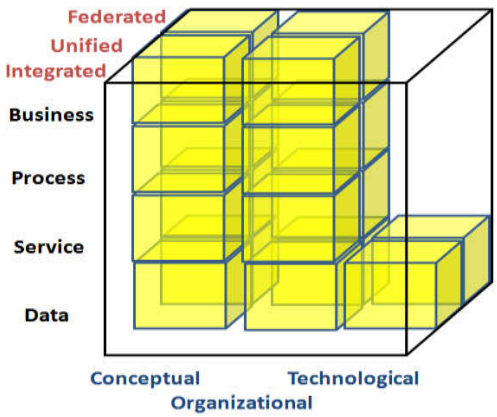

Figure 1. EOS positioned in the interoperability framework.

The execution of EOS has been designed to respect the most recent interoperability approaches [13] to help the improvement of federated and unified methodologies of enterprise interoperability. This respect of interoperability is set from conceptual level down to technical level, where as one of the first implementations, the High-Level Architecture (HLA) [14] has been utilized to quickly create an HLA-based interface for achieving federated enterprise interoperability [15].

Further, interoperability is becoming increasingly critical, but paradoxically, it is not yet fully anticipated, monitored, and accompanied in an effective way to recover from incompatibility problems or failures. [16] stated that enterprise modeling, enterprise interoperability, and model-driven approaches can contribute, along with systems engineering architecture to develop and improve interoperability in enterprise systems. The Model-Driven Systems Engineering Architecture (MDSEA), extended to interoperability problems, led to the design of the Model-Driven Interoperability System Engineering (MDISE) framework, which leverages enterprise interoperability research and can be used in the transition from EOS concepts to GreneOS operationalization.

3. EOS to GreneOS

To show the position of the EOS regarding enterprise architecture layers, this section aligns it with the ArchiMate Core framework**.** The ArchiMate framework is a reference structure used to classify elements and allowing to follow the modeling of the enterprise from different viewpoints, where the position within the cells highlights the concerns of the stakeholder. It consists of three layers (Business, Application, and Technology) and three aspects (Active, Behavior, and Passive). Each cell of the framework makes available a set of modeling objects allowing to describe a specific aspect of an enterprise architecture. The work proposed in this article is situated above the three aspects and three layers, as shown in Figure 2. The ArchiMate Core Framework is guiding and paving the way according to the MDSEA-model-driven approach developed in this project. The following figure shows the positioning of the EOS and GreneOS regarding to the ArchiMate layers.

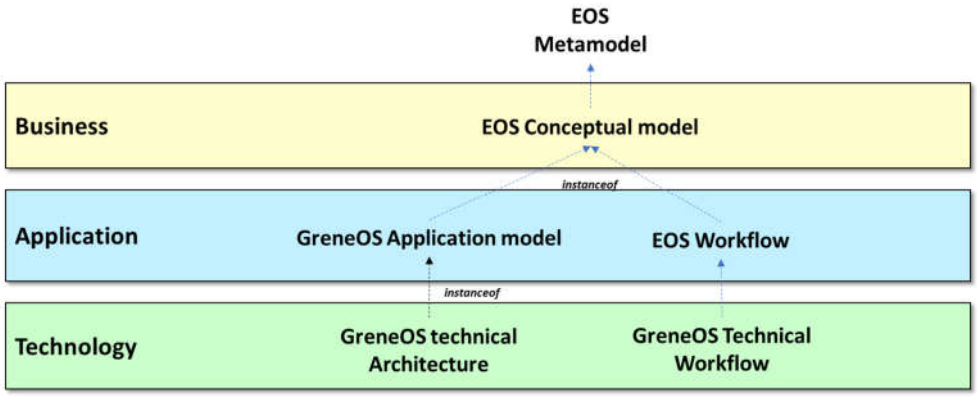

Figure 2. EOS within the ArchiMate Core framework.

3.1. EOS Meta Model to Conceptual Architectures

This work is based primarily on the EOS metamodel, which presents the basic objects that inspired the development of the EOS architecture and GreneOS. The EOS metamodel highlights the concepts and relationships that need to be instantiated to develop a model at the business level of the enterprise architecture, then derives this model at the application level, then at the technical level, and finally at the solution implementation. As described in Figure 3, it contains the following concepts: Enterprise Resource Management (ERM), Enterprise Process Management (EPM), Enterprise Information Management (EIM), Presentation Management (PM), and Interoperability Management (IM).

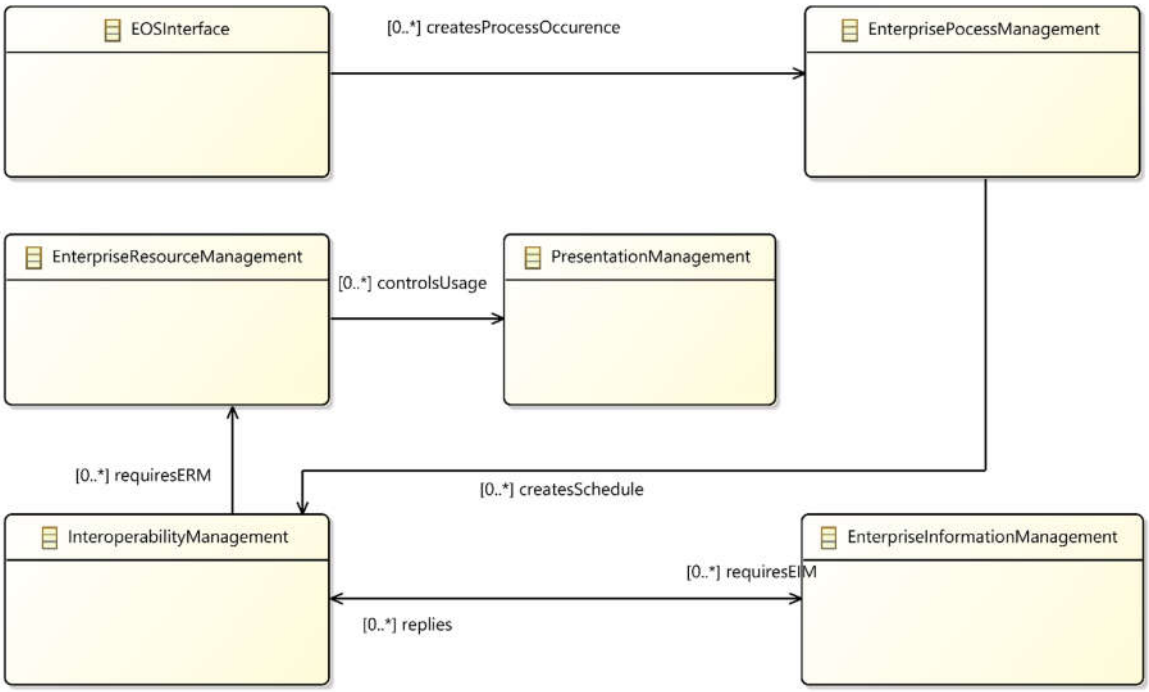

Figure 3. EOS metamodel.

3.2. Conceptual Architecture

In previous works, the EOS conceptual architecture has been utilized as a framework-wide stage that enables the business managers to interface with and convey through the enterprise systems (hardware, software, network, machines, human operators...) in a powerful and proficient way [2,17].

As shown in the Figure 4, three types of resources have been considered as the components of the Internet-of-Everything, the business managers, and the Interoperability Interface are outside the EOS. They are associated with the EOS to send and get directions (or requests) and data (for example information, inputs, statuses...). To perform endeavor activities (i.e., to design, manufacture and sell required products to customers) in an interoperable and composed way, the assets must be commanded and controlled by business supervisors by means of the EOS.

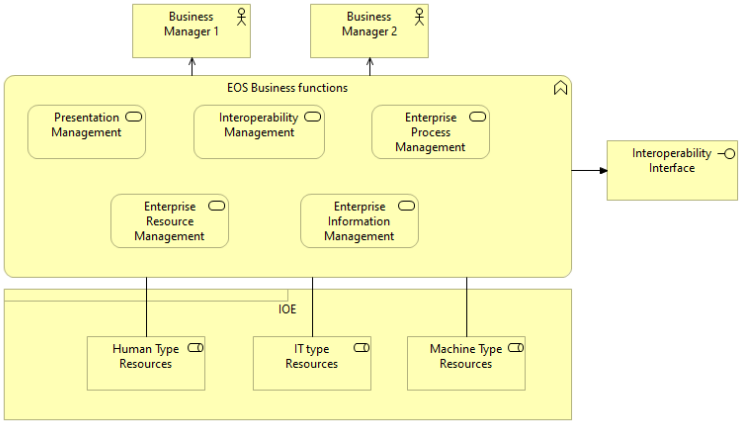

Figure 4. EOS business architecture.

According to Figure 4, the EOS business architecture is made of five services to guarantee the working of the EOS:

- Enterprise resource management dynamically monitors the status of enterprise resources, allocates suitable resources to executing operations, and guarantees that the right resources are assigned to the perfect place at the ideal time. It tackles the vision that consists of having a smooth integration of the physical (human, machine), so CPS with the information system is orchestrated by the EOS.
- Enterprise process management executes business forms characterized by business clients, sends directions setting off the beginning of procedures, records finishing status of activities (done, fail, not done...), and executes the EOS inner procedures/activities.
- Enterprise information management oversees, secures, and supports data and data exchange of different sorts between the enterprise's resources associated with the EOS, and guarantees data and information classification and security to shield from nonapproved access.
- Presentation management is a set of services with fitting interfaces that permit business clients and other enterprise resources to associate with the EOS and get/send data.
- Interoperability management is a set of services that provides basic mapping between heterogeneous assets to make them interoperable through the EOS [1].

3.3. EOS Implementation Architecture

Implementation architecture describes how to physically realize a framework. It points out and characterizes the technical innovations and concepts chosen to execute the technical architecture. In other words, the implementation architecture identifies the needed technological components and their interactions to build a system.

Initially, the High-Level Architecture (HLA) [14] has been utilized to quickly create an HLA-based interface for achieving federated enterprise interoperability [15].

Broadly speaking, for a given technical architecture, different diverse advancements can be utilized for the execution. In the proposition of [18], the implementation of the HLA with a Run-Time Infrastructure and federates was chosen [10,19]. The fundamental explanations behind this decision were (1) its capability to effortlessly empower federated interoperability between heterogeneous applications and (2) its ability to manage expansive distributed (simulation) environments. Plus, the HLA is likewise in accordance with service orientation-based implementation that is one of the primary attributes of EOS standards. Indeed, an EOS is mostly an IT framework made of numerous services. The HLA gives one conceivable implementation support particularly convenient to the EOS. Nevertheless, the implementation of the EOS is not restricted to the HLA [20].

All the EOS federates (EOS components) were associated together through an HLA Federation as to publish and subscribe events between federates. This kind of structure was utilized to facilitate interfederate communications and to help productive data exchange between EOS components when participating in a distributed federation execution [21].

The HLA technology permits the EOS federates to convey information and to synchronize activities among each other, dependent on the federated interoperability features, by characterizing how federates can associate with the RTI, create, join, and manage federations, save, and re-establish federation states and characterize a framework to synchronize federates to a similar time [22].

3.4. GreneOS Vision

3.4.1. GreneOS Application Architecture

GreneOS follows the EOS philosophy by eliminating the need to use multiple applications for teams of users and processes replacing most of this disparate

applications/software with one unified platform. GreneOS can be considered as one of the most recent implementations of the EOS metamodel. GreneOS connects all employees, customers, vendors, drones, machinery, robots, facilities, and third-party resources to one single platform, making sure that 100% of an enterprise's internal and external resources converse on a single unified platform. This results in a manifold increase in efficiency, transparency, and actionable business intelligence in real-time, launching an end-to-end transformation where GreneOS itself becomes the enterprise's value chain driver.

Figure 5 presents the poles that build GreneOS at the application level. The viewpoint of GreneOS is that decisions are as good as your data. As described in Figure 5, GreneOS intends to bring 100% of the enterprise workforce (human and robotic) onto the platform from the delivery executive to the CEO, with no limit on the number of users and a transaction-based pricing model. GreneOS is an intelligent AI-powered platform that continuously learns and becomes smarter each day with machine learning, in turn making your enterprise hyperefficient and more productive (expect 10 times of × increase in efficiency).

As recommended by the EOS, GreneOS is not a one-size-fits-all application. It is customized and clinically put together for each customer to seamlessly fit into their culture, needs, and challenges. We do not believe in cookie-cutter solutions. Unlike traditional enterprise software, GreneOS can be rapidly deployed within 6–8 weeks thanks to our transformational zero-code development environment that uses a drag-and-drop interface. Once the system is implemented, AI enables it to continuously learn and evolve with every data input and customer feedback, while increasing efficiency and cutting downtime taken to make decisions.

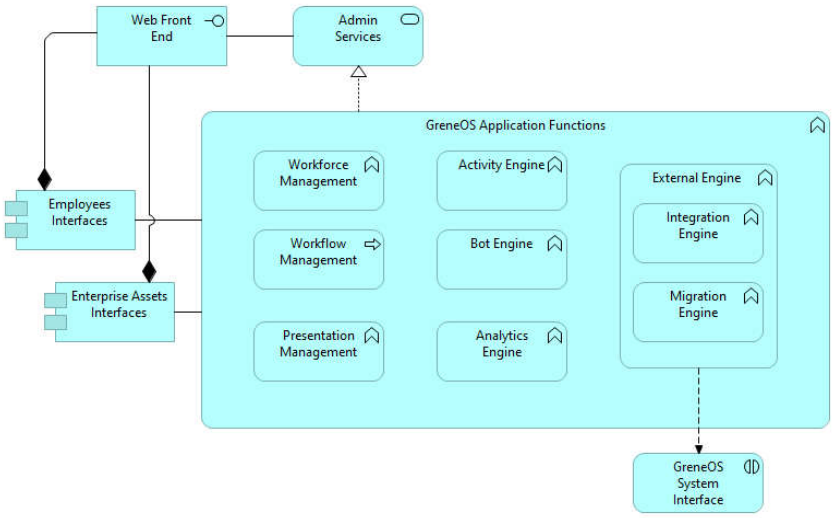

Figure 5. GreneOS application architecture.

3.4.2. EOS Business to GreneOS Application Architecture

The EOS provides a general framework, meaning that all the features do not need to be systematically implemented in the concrete system that is derived from it. Nevertheless, GreneOS gives a good covering of the EOS functions, as described in Figure 6, even if both approaches derive from the same metamodel. Figure 6 illustrates the transposition from the EOS to GreneOS architectures based on the functionalities, features, and services of each component:

- Business Managers and Admins of both the EOS and GreneOS are not monitored and controlled by the platforms; they are the decision-makers who define what and how enterprise operations will be done, manage the activities, and send commands to enterprise resources via the EOS and GreneOS.

- Human, machine, and IT resources connected to the EOS are defined as Employees and Assets for the GreneOS; they are defined as the sets of individuals, machines, and computer elements that are interacting with the EOS and GreneOS.
- Both ERM component of the EOS and the Workforce/Workflow Management component of GreneOS provide a real-time and global view of the 'occupation' of the resources in a company by monitoring the enterprise resources system-wide (available, occupied, out-of-order…).
- The EPM component of the EOS, same as the Activity/Bot/Analytics Engines of GreneOS, are responsible for interpreting process behavioral, information, and resource requirements.
- The Presentation Management module set services are the same for both approaches; this component with the appropriate interfaces allows business users and other enterprise resources to connect to the EOS and GreneOS and receive/send information.
- The Interoperability Management component of the EOS plays the same role as the Integration/Migration Engines of the GreneOS; they are presented as sets of features that provide necessary mapping between heterogeneous resources to make them interoperable through the EOS and GreneOS.

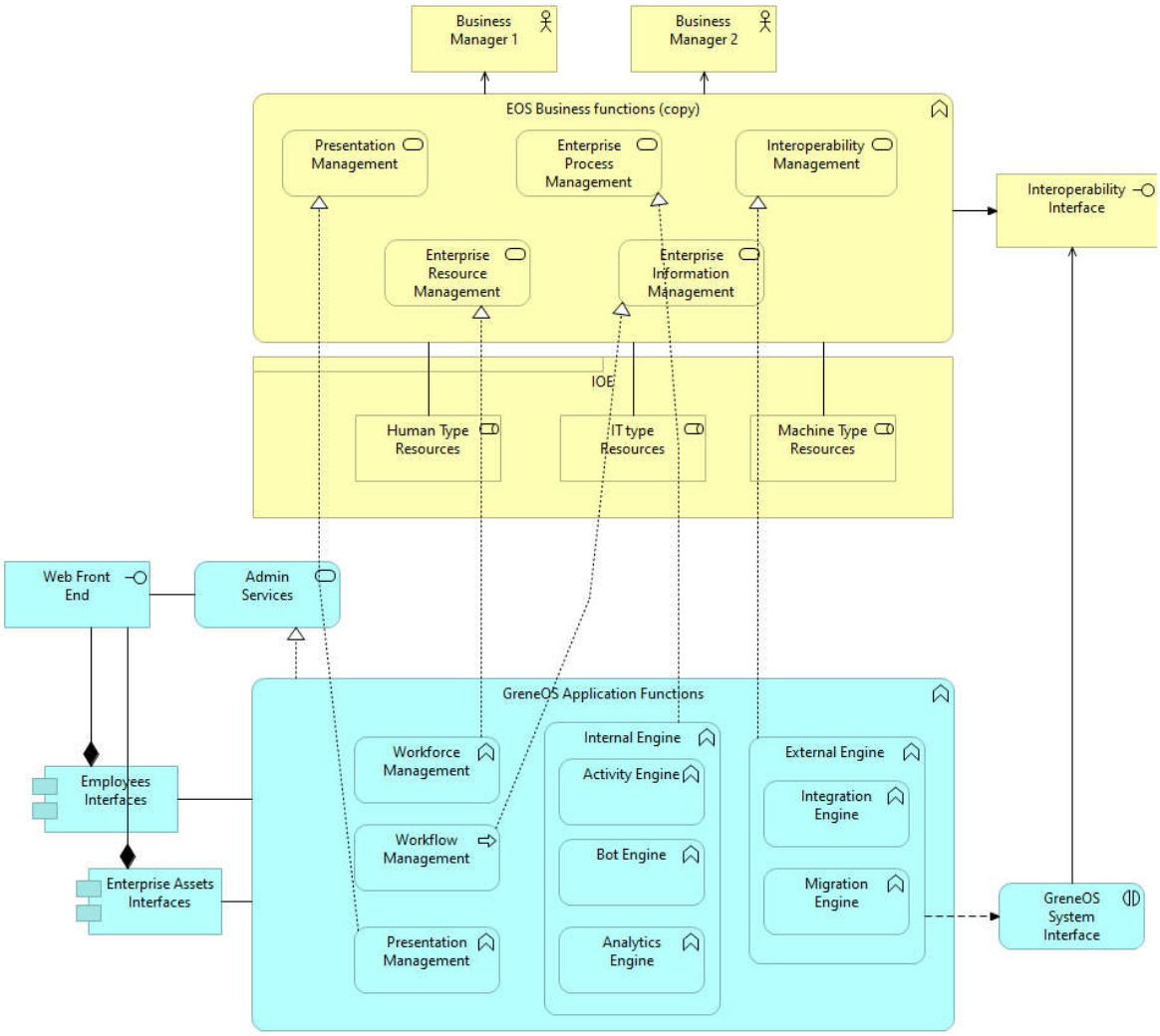

Figure 6. Alignment of EOS business architecture with GreneOS application architecture.

Figure 7 illustrates and presents an EOS application layer as workflow where the enterprise operations are executed and generated through the EOS internal components [23], noting that this architecture is designed using the HLA technology during the year 2017. The final objective is executing this workflow within the GreneOS technical architecture.

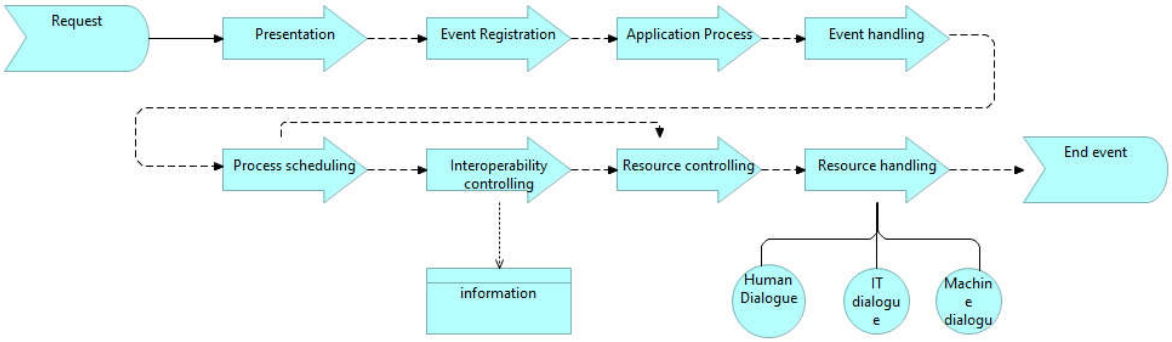

Figure 7. EOS workflow at application layer.

At first, each decision-maker accesses the General-Purpose and Vertical software interfaces to request a new job from the day-to-day list of activities and operations. The software sends related commands and data to communicate with the EOS front-end interface named Presentation Module to execute the requested job.

The Presentation Module then interprets the run-time entities, creates the events which will be registered with their associated information by the Event Registration Component using the Run-Time Repository Service, and produces event occurrences for the Event Handling component.

Next, the Event Handling deals with the event occurrence needs, lines, and traceability, and gives the request identifier and makes the Process Occurrence. The Process Occurrence requests planning from the Process Scheduling part, which deciphers the process behavior, data, and resource necessities. This subservice checks the approval to execute the process, recovers process depictions from the Run-Time Repository, summons the Enterprise Resource Management to allocate the needed resource capabilities, and advances the details to the Rule Interpretation segment.

Afterward, the Rule Interpretation segment provides usefulness to recover the sequencing and contingent principles related with the distinguished Enterprise Process, saves a state record of all enterprise activities, and reacts to identified events to instate or end the activity.

In this manner, the Activity Occurrence plan made by the Rule Interpretation is sent to the Interoperability Component, which is trustworthy and asks the Enterprise Resource Management to appoint assets apportioned by the Process Scheduling Component, conjuring the Enterprise Information Management to acquire the item states and demonstrate the information object required, asking the Enterprise Resource Management to: assign resources designated by the Process Scheduling segment, discharge the involved resources, while ending an activity and notifying the termination of the present action to the Rule Interpretation component [13].

The Resource Controlling part checks the accessibility of the resources and preallots them when accessible, reacts to the Interoperability Controlling solicitations to relegate specialists, and reacts to the Process Scheduling requests to designate and deallocate resources.

The Resource Handling segment guarantees asset streamlining and resource mapping by choosing the proper resources after coordinating the abilities required and by thinking about the time, performance, and priority.

The Presentation Management services are constrained by the Resource Handling Component for dealing with Human, Machine, and IT Dialogs:

- The Human Dialog provides usefulness by introducing in the fitting configuration the present status and the previous history of occasions, enabling authorized persons to mediate manually to adjust the relevant parameters at run-time.
- The Machine Dialog bolsters the vital features to give access to the different functional capabilities of the machine. It provides the usefulness required to receive and decipher responses from the machine.
- The IT Dialog gives functionality to the cross-examining application program interfaces to decide its capacities, offering help for the joining of the practical substances executed by existing IT application programs.

3.4.3. GreneOS Technical Architecture

One of the core aspects of GreneOS is the unification of all resources in an enterprise. The resources here could include humans and any other IoT based devices that are part of the enterprise. Hence, it was established that we will have to deal with huge volumes of data when data are being captured from the entire spectrum of resources.

To handle such huge volumes of data, we have gone ahead with a decoupled data stream architecture described in Figure 8. As you can see in this diagram, the data requests are not immediately processed. First, the data are produced on to the messaging system and immediately an acknowledgement is sent to the client applications stating that the request has been received. On the other side of messaging systems, we have consumer engines that process the request data asynchronously. This enables the system to handle higher loads with the same amount of computation power, as the connections with the clients are not persisted until the processing of the data is completed.

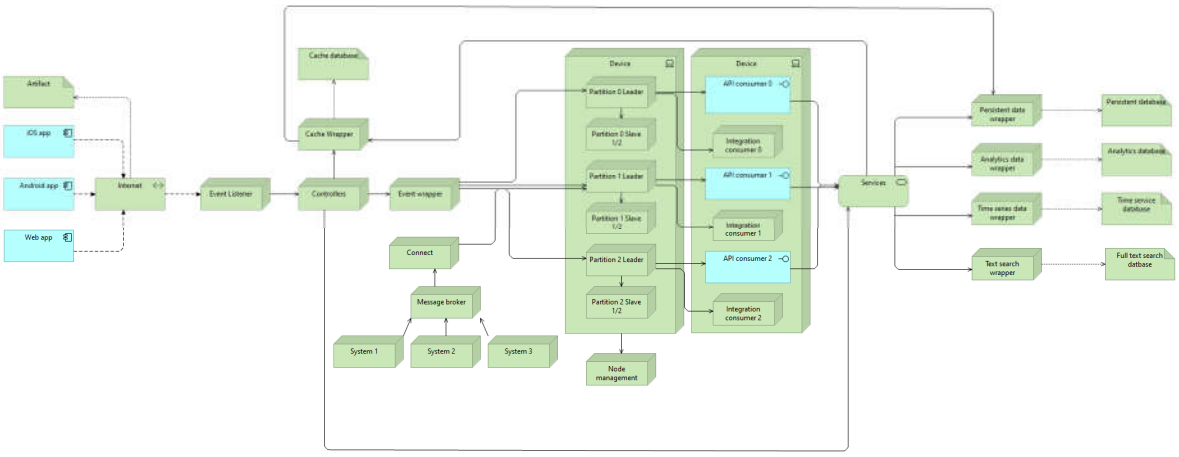

Figure 8. GreneOS Technical Architecture.

Another important aspect of this architecture is that all the server-side components are loosely coupled to the data stream. Especially, every storage application is abstracted by the presence of a wrapper code. This essentially builds a lot of futureproofing into the architecture, as it will be easy to replace a specific server component when there are more competent technologies available.

Now, let us look at the typical data flow. Client apps send requests, which are received by controllers on the server side. Upon receiving requests, the controllers convert the request to a standard message format while incorporating the required metadata to identify the endpoint of the request. Converted messages are then produced to the messaging system. Consumers retrieve the produced messages on the other side of the messaging system. The consumer identifies the corresponding service component by

retrieving the endpoint from the metadata. The consumer then invokes the corresponding service for the message. The service processes the data based on the specific use case copies and transforms data into one or multiple storage applications. Please note that only write requests leverage the decoupled data stream architecture. For all the read requests, the messaging system is bypassed and the service is directly invoked.

4. Use Case

4.1 EOS Workflow to Orchestrate GreneOS Technical Architectures

The matching between both technical architectures (GreneOS and EOS) is based on the functionalities, features, and services of each component.

The General-Purpose Vertical Software of the EOS and the core applications (Media, iOS, Android, Web) of GreneOS play the same role as computing resources that request a new job from the day-to-day list of activities and operations. The software sends related commands and data to communicate with the EOS or GreneOS

The ERM module of the EOS containing the Resource Handling and Resource Controlling components is matched with the Ordered Messaging System of GreneOS; they have the role to check the availability of the resources and preassign them when available, assign agents, and ensure the resource optimization and the resource mapping by selecting the appropriate resources after matching the capabilities required and by taking into consideration the time, performance, and priority

Both the EIM component (Information Library and Object) of the EOS and the Persistent/Analytics/Time/Search Databases of GreneOS must manage the exchanged messages and information between resources, acquire the object states, and specify the information object required

The IM component of the EOS plays the same role as the Consumer Group of GreneOS: it requests from the Resource Management to assign the specified resources/employees/assets and invokes the EIM/Databases to acquire the information and data.

Use-case EOS Workflow with GreneOS

GreneOS can handle a workflow process initiated from a discussion with a chatbot presenting different journey options of resources users. The use-case example is described in Figure 9.

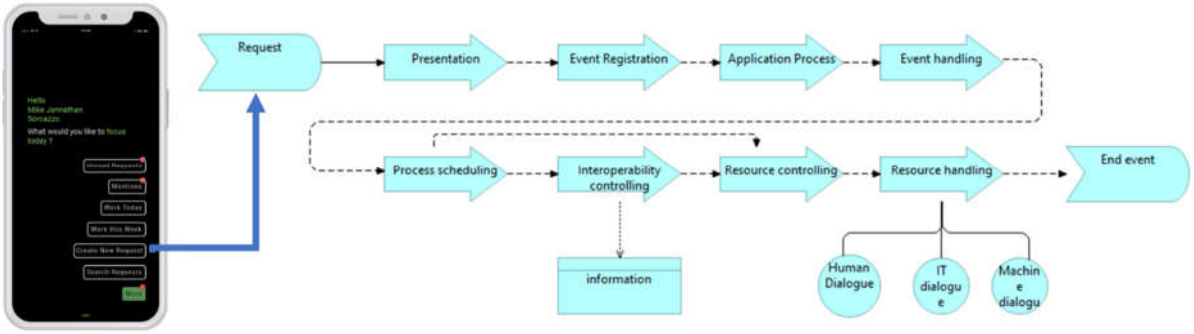

Figure 9. GreneOS tasks management in a chat.

The implementation of "Business Manager requires a new job" is being detailed with the interfaces of GreneOS. Figure 9 presents how the user chat interface on the smartphone with the workflow part.

It starts from the client apps that send requests which are received by controllers on the server-side. Upon receiving requests, the controllers convert the request to a standard message format while incorporating the required metadata to identify the endpoint of the request. Then, the converted messages are produced to the messaging system. At that point, according to Figure 8 descriptions, consumers retrieve the produced messages on the other side of the messaging system. At that time, the consumer identifies the corresponding service component

by retrieving the endpoint from the metadata. The consumer then invokes the corresponding service for the message. From the service component, the following modules will be invoked based on the use case—Bot Engine, Analytics Engine, and Integration Engine. The Bot Engine drives all the autonomy that is relevant for the current data request type. Likewise, the Analytics Engine is fired to crunch the data required to drive the dashboards and reports, and the Integration engine is fired to exchange data with the partner systems. Finally, postprocessing of the data based on the specific use-case copies will have transformed the data into one or multiple storage applications.

5. Discussion and Perspective

5.1. Workflow from WFMC

Founded in 1993, the Workflow Management Coalition (WfMC) is a global organization based on diverse groups of workflow and BPM contributors. Based on WfMC works, [24] previously announced that there is a crucial need for a workflow-explicit orchestration editor revealing and proposing to edit the most frequently hidden processes of the information. Yet, few ERPs are including an explicit workflow editor, making the user tempted to adapt his process more easily to the one defined in the ERP while it should be the ERP that should adapt. The EOS has a role to play in the proposition of workflow editors and transformation into executable processes.

From another perspective, process discovery (including process mining [25] with the reuse of legacy processes and the mapping of existing processes may be of interest to enhance the EOS with an artificial intelligence function to define processes in an automated or guided manner. One can imagine an adaptive workflow progressively built from the chat option chosen in the discussion.

5.2. ESB and SOA

The Enterprise Service Bus (ESB) has been defined since [6,26] as a set of tools that guarantee the confidence of data exchanges between the sources and targets of an information system. Thus, it is implemented to allow heterogeneous applications to connect (connectivity) and share/exchange information (interoperability) through a software architecture. Recent works [27] show that the ESB methods, approaches, and technologies have now reached a good maturity of functionality in combination with selected tools to achieve a good integration of information systems. A recent systematic literature study [28] explained that ESB remains the best promising way for the integration of business applications in distributed and diverse environments. Nevertheless, ESB works by a data exchange structure that is mostly standardized and, rather than making direct interfaces between applications, these applications subscribe to the messages available in the ESB that interest them [29].

In the frame of the EOS, the main component of an ESB is still a route that allows information to circulate between applications (an "application bus") and whose purpose is to guarantee the routing of data as well as its persistence [30]. However, in addition, we assume that the bus connection can be achieved with connectors (interoperability with legacy [31] implemented with low code obtained again from discussion with users. So, the EOS supports data transport in a more flexible way, being supported by ontologies and semantics with the ability to route and deliver large volumes of data in real time more easily.

The ESB is a part of SOA for business integration, which is an avenue of research. The research proposed by [32] addresses this integration by proposing a conceptual SOA model for achieving different levels of enterprise integration.

5.3. Graphical Editor and Visualization Tools

The workflow can be built on the fly upon the discussion between the chatbots and users thanks to modeling and simulation tools. We can assume that model animation can guide the user in the setting of the workflow before to launch it. The heat map is now frequently used in the domain of process modeling and can be transposed to the EOS.

6. Conclusions

This article started from the observation that current ERP systems are still limited by customization, implementation time, and interoperability with other systems.

A first contribution of the article is the detailed explanation, using the ArchiMate formalism, of the conceptual architecture, the technical architecture, and the implementation architecture of the EOS following the previous conceptual contributions about the EOS. The paper also shows how the EOS is guided by interoperability in the MDSEA framework. Therefore, it demonstrates that the EOS can be used to address the services and functionalities needed for Enterprise 4.0.

To go further, this conceptual architecture with the basic functions proposed in the EOS, as well as the technical architecture were applied to GreneOS, which is one of the first operational implementation architectures benefiting from the innovation and concept of the EOS. This demonstrated a successful implementation in a robust technical architecture driven by the EOS concepts.

The article therefore discusses the current interest of EOS with an application case, showing the complementarity and the limits of the conceptual architecture of the EOS and its implementation in GreneOS. It finally concludes by proposing perspectives for future developments of the EOS.

Author Contributions: Conceptualization, J.R., B.M., N.D. and G.Z.; methodology, J.R., B.M., N.D. and G.Z.; software, J.R., B.M., N.D. and G.Z.; validation, J.R., B.M., N.D. and G.Z.; formal analysis, J.R., B.M., N.D. and G.Z.; investigation, J.R., B.M., N.D. and G.Z.; resources, N.D., J.R, B.M. and G.Z.; data curation, N.D., J.R, B.M. and G.Z.; writing—original draft preparation, J.R., B.M., N.D. and G.Z.; writing—review and editing, J.R., B.M., N.D. and G.Z.; visualization, J.R., B.M., N.D. and G.Z.; supervision, J.R., B.M., N.D. and G.Z.; project administration, J.R., B.M., N.D. and G.Z.; funding acquisition, J.R., B.M., N.D. and G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data has been presented in main text.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Youssef, J.R., Zacharewicz, G., Chen, D., Vernadat F. EOS: Enterprise Operating Systems. *Int. J. Prod. Res.* **2018**, *56*, 2714–2732
2. Youssef, J.R., Chen, D., Zacharewicz, G., Zhiying T. Developing and Enterprise Operating System (EOS) with the Federated Interoperability Approach. In Proceedings of the I3M'16—International Multidisciplinary Modeling Simulation Multiconference, Larnaca, Cyprus, 26-28 September, 2016
3. Youssef, J.R., Zacharewicz, G., Chen, D. Developing an Enterprise Operating System (EOS)–Requirements and Architecture. In Proceedings of the 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises—WETICE-2016, Paris, France, 13–15 June 2016.
4. Burnson, F. *Enterprise Resource Planning Software—Buyer Report*; Software AdviceT™: Stamford, England, 2015.
5. Ahmad, N., Naveed, Q. N., Hoda, N. Strategy and procedures for Migration to the Cloud Computing 2018. In Proceedings of 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences, Bangkok, Thailand, 22–23 November 2018.
6. Guerola-Navarro, V., Gil-Gomez, H., Oltra-Badenes, R., & Sendra-García, J. Customer relationship management and its impact on innovation: A *Lit. Rev.* **2021** *129*, 83–87.
7. Chang, S., Gable, G., Smythe, E., Timbrell, G.A. Delphi examination of public sector ERP implementation issues. In Proceedings of the International Conference on Information Systems, Brisbane, Australia, 10–13 December 2000
8. Connecza. ERP solution for Educational Organizations|Institutional ERP System. Available online: https://www.wattpad.com/amp/380694632 (accessed on 28 September 2022).
9. Shankararaman, V,, Kit, L.E. Integration of Social Media Technologies with ERP: A Prototype Implementation. In proceedings of AMCIS 2013, Chicago, Illinois, USA, 15–17 August 2013,

10. IEEE . IEEE Standard for Distributed Interactive Simulation–Application Protocols. Available online: https://standards.ieee.org/ieee/1278.1/4949/ (accessed on 28 September 2022).
11. Fitriyani, F., Rosadi, K. I., & Alfian, M. Literature Review Determination of Planning and Organization in Educational Institutions: Analysis of Leadership and Organizational Culture 2020, *Dinasti Int. J. Digit. Bus.Manag.* 2022, 3(4): 584-592.
12. ISO/IEC. ISO/IEC 10746:2009, Information Technology – Open Distributed Processing – Reference Model: Architecture 2009 Available online: https://www.iso.org/standard/55724.html (accessed on 28 September 2022).
13. Chen, D., Youssef, J. R., Zacharewicz, G. Towards an Enterprise Operating System—Requirements for Standardisation. In Proceedings of the IWEI Workshops, Nîmes, France, 27 May 2015.
14. Fujimoto, R. The High Level Architecture: Introduction. In Proceedings of the First International Workshop on Distributed Interactive Simulation and Real Time Applications, Eilat, Israel, 9–10 January 1997.
15. Salinesi, C., Norrie, M. C., Pastor, Ó. *Advanced Information Systems Engineering*. In proceedings of the CAiSE 2013, Valencia, Spain, June 17-21, 2013.
16. Zacharewicz, G., Bazoun, H., Ribault, J., Boyer, H. SLMToolBox: Enterprise service process modelling and simulation by coupling DEVS and services workflow. *Int. J. Simul. Process Model.* **2016**, *11*, 453–467. https://doi.org/10.1504/IJSPM.2016.10001547.
17. CEITON. "Front-end and back-end EAI", CEITON technologies. Wikipedia.2014. Available online: https://en.wikipedia.org/wiki/Enterprise_application_integration (accessed on 28 September 2022).
18. Youssef, J., Zacharewicz, G. *E*nterprise Operating System (EOS) in action: Distributed Simulation of Enterprise Activities and Operations. In Proceedings of the 2019 Winter Simulation Conference, National Harbor, MD, USA, 8–11December 2019.
19. Portico. Portico Space : www.porticoproject.org 2013. Available online: https://portico.space/youre-the-architect (accessed on 28 September 2022).
20. Weichhart, G., Molina, A., Chen, D., Whitman, L.E., Vernadat, F. Challenges and current developments for sensing, smart and sustainable enterprise systems. *Comput.Ind.*, **2016**, *79*, 34-46.
21. Zhiying Tu. Federated Approach for Enterprise Interoperability: A Reversible Model Driven and HLA Based Methodology. Available online: https://www.theses.fr/2012BOR14673 (accessed on 28 September 2022)
22. Pokorny, T. Portico 2016. Available online: https://sourceforge.net/projects/portico/?source=recommended (accessed on 28 September 2022).
23. Youssef, J.R., Chen, D., Zacharewicz, G. Developing and Enterprise Operating System (EOS)-State of the Art. In Proceedings of the ICE/IEEE'17—International Conference on Engineering, Technology and Innovation, Antalya, Turkey, 21–23 August 2017.
24. Zacharewicz, G., Chen, D., Vallespir, B. HLA Supported Federation Oriented Enterprise Interoperability, Application to Aerospace Enterprises. In Proceedings of 2008 EURO International Simulation Multi-conference, Edinburgh, Scotland, 16–19 June 2008.
25. Van Der Aalst W. Process mining: Overview and opportunities. *ACM Trans. Manag. Inf. Systems.* **2012**, *3*, 1–17.
26. Schmidt, M.T., Hutchison, B., Lambros, P., Phippen, R. The enterprise service bus: making service-oriented architecture real. *IBM Syst.J.* **2005**, *44*, 781–797.
27. Shkuta, V., Miłosz, M. Comparative analysis of tools for the integration of IT systems. *J. Comput. Sci. Inst.* **2021**, *18*, 30–36.
28. Aziz, O., Farooq, M. S., Abid, A., Saher, R., Aslam, N. Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study. *IEEE Access* **2020**, *8,* 31180–31197.
29. Turner, K.J. Advances in Home Care Technologies, Results of the MATCH Project. IOS Press: Amsterdam, The Netherlands, 2012.
30. Zurawski, R. *The Industrial Information Technology Handbook*; CRC Press Book: Boca Raton, FL, USA, 2004.
31. Chen, D., Daclin, N. Framework for enterprise interoperability. In *Interoperability for Enterprise Software and Applications*; Wiley: London, UK, 2006.
32. Grant, D., Yeo, B. Enterprise integration using Service-Oriented Architecture. *Issues Inf. Syst.* **2021**, *22,* 164–177.