Article

Diversification of Legislation Editing Open Software (LEOS) Using Software Agents—Transforming Parliamentary Control of the Hellenic Parliament into Big Open Legal Data

Sotiris Leventis [1], Fotios Fitsilis [2,*] and Vasileios Anastasiou [3]

[1] Hypernetica, 15125 Marousi, Greece; sotiris.leventis@hypernetica.com
[2] Scientific Service, Hellenic Parliament, 10671 Athens, Greece
[3] Hellenic OCR Team, 15343 Agia Paraskevi, Greece; info@hellenicOCRteam.gr
* Correspondence: fitsilisf@parliament.gr; Tel.: +30-6947818439

Abstract: The accessibility and reuse of legal data is paramount for promoting transparency, accountability and, ultimately, trust towards governance institutions. The aggregation of structured and semi-structured legal data inevitably leads to the big data realm and a series of challenges for the generation, handling, and analysis of large datasets. When it comes to data generation, LEOS represents a legal informatics tool that is maturing quickly. Now in its third release, it effectively supports the drafting of legal documents using Akoma Ntoso compatible schemes. However, the tool, originally developed for cooperative legislative drafting, can be repurposed to draft parliamentary control documents. This is achieved through the use of actor-oriented software components, referred to as software agents, which enable system interoperability by interlinking the text editing system with parliamentary control datasets. A validated corpus of written questions from the Hellenic Parliament is used to evaluate the feasibility of the endeavour, and the feasibility of using it as an authoring tool for written parliamentary questions and generation of standardised, open, legislative data. Systemic integration not only proves the tool's versatility, but also opens up new grounds in interoperability between formerly unrelated legal systems and data sources.

Keywords: parliamentary control; LEOS; Akoma Ntoso; software agents; multi agent systems; enterprise integration patterns; system integration; design patterns; big open legal data; interoperability

Citation: Leventis, S.; Fitsilis, F.; Anastasiou, V. Diversification of Legislation Editing Open Software (LEOS) Using Software Agents—Transforming Parliamentary Control of the Hellenic Parliament into Big Open Legal Data. *Big Data Cogn. Comput.* **2021**, *5*, 45. https://doi.org/10.3390/bdcc5030045

Academic Editor: Min Chen

Received: 18 July 2021
Accepted: 16 September 2021
Published: 18 September 2021

1. Introduction

1.1. Background

Legal informatics, a core element of government technology (GovTech) and legal technology (LegalTech), is on the rise and has the potential to transform entire parts of the public sector [1]. As such, it is a broad interdisciplinary field at the crossroads between law and information science, with particular links to the semantic web [2]. The semantic web is a web of linked data that has been enabled through a series of standards and technologies, such as the resource description framework (RDF) and web ontology language (OWL), which again builds on XML (extensible markup language), a metalanguage allowing transformation of documents into machine-readable content [3]. The above standards provide semantic characteristics, interlink, and apply rules for handling an ever-increasing distributed data volume.

Parliaments, which as organisations do not exactly have a reputation of being technology affine or, at best, displaying a cautious approach against information and communication technologies (ICT) [4], may benefit from the development of a wide range of legal informatics tools and services [5]. The output of parliamentary procedures tangibly appears in the form of legal documents in various forms and formats. While the above technologies can certainly be applied to digitise parliamentary workflows and documents, it was not until recently with the standardisation of Akoma Ntoso [6], an XML-based markup scheme specifically designed for the semantic representation of legal documents, such as proceedings, bills and means of parliamentary oversight, that made the design of parliamentary ICT systems a mainstream endeavour. The development of Akoma Ntoso has been developed under the United Nations Department of Economic and Social Affairs (UN/DESA) programme 'Africa i-Parliament Action Plan'. In August 2018, it was accepted as OASIS standard (Version 1.0).

At the central European level, this led to the development of Akoma Ntoso (AKN) for European Union (AKN4EU) [7], a data scheme for the representation of EU legal documents as well as for a series of open-source software tools, such as the Authoring Tool for Amendments [8], which is used at the European Parliament, and Legislation Editing Open Software [9], a web authoring tool designed for collaboratory legislative drafting. Both tools natively support AKN document formats. Despite their growing importance, particularly in the European area, assessment of these tools is scarce, partial and inconclusive [10,11].

1.2. Transformation of Parliamentary Data

From an organisation's point of view, despite support by EU institutions, a lack of evaluation data proves to be a rather significant drawback for an eventual adoption of such tools. This has been the case in the study of the parliamentary control function by the Hellenic OCR Team [12], a scientific crowdsourcing initiative for the processing and analysis of parliamentary data. The Hellenic OCR Team is an open research platform and a parliamentary spin-off. It links experts from academia and the public and private sectors who jointly tackle administrative and operational issues, as well as strategic challenges faced by parliaments with scientific methods and precision. While the team specialises in the transformation of existing parliamentary data to (linked) open data, to date there is no integrated tool for members of parliament (MPs) for the authoring of new parliamentary control documents (for instance, written parliamentary questions).

Parliamentary functions, such as law-making and oversight, are supported by procedures that generate huge numbers of legal documents. The legal data contained therein, in machine-readable format and from possibly multiple sources, has the potential to unlock the governance black-box for citizens and other public stakeholders, thus promoting democratic accountability and the rule-of-law [13,14]. The development of legal informatics systems and platforms for authoring and handling of aggregated, complex legal data has recently been the focus of dedicated studies [15,16]. The sheer volume and variety of available data sources that can potentially be accessed and inter-linked through the approach taken in this article (that is, the building of dedicated software agents using Enterprise Integration Patterns) collectively qualifies them as big data. For the aforementioned structured, hierarchical, metadata rich objects, the term Big Open Legal Data (also known through the acronym BOLD) has been coined. As big data continue to grow, their broader benefit and societal impact can be substantial [17]. For instance, big data can fuel artificial intelligence assisted decision and policy-making systems [18]. From the multitude of potential applications of artificial intelligence (AI) in the legal data analytics domain, one may here indicatively mention the following as particularly relevant for the intra-parliamentary workspace: legal reasoning; compliance checking; information clustering; and classification. With analysis, data becomes information. Yet, as demonstrated by Devins et al., extracting information out of big data can be problematic in the legal context [19].

This is due to the fact that despite advances in the fields of big data and artificial intelligence, machines are not yet in a position to replace the ability of legal professionals to interpret legal data based on naturally non-definitive and current or changing contexts and complex and diverse perspectives. This is due to legal data being subject to multiple and indefinite meanings and, thus, interpretations. The use of big data could potentially and partially tackle this problem, but could also lead to self-reinforcing feedback loops, thus locking into local equilibrium positions [19]. The authors believe that a framework has the potential to tackle the above risks and challenges if it brings together the current ability and potential of suitable technologies with the value of human judgment and expertise, both individually and collectively.

Such a framework should approach the challenges by providing appropriate tools to legal professionals, in terms of assisting them in achieving their objectives at hand. Furthermore, a modular and distributed architecture would encourage collaboration in the domains of legal informatics, big data, and artificial intelligence. These assistive technologies could take the form of software components that can supplement human shortcomings, such as the ability to retrieve and identify relevant information and enhance its strengths, such as decision-making, through recommendations. These components need to be able to interact and communicate with each other, as well as with other systems, while interacting and assisting their human counterparts.

1.3. Objectives and Methodology

The article presents a framework where proven enterprise integration pattern (EIP) techniques are used [18,20,21] that are utilised by software agents in a manner that can be horizontally scaled to permit the processing of indefinitely large data sources and independently of data definitions. Kurth provides an extensive annotated bibliography of articles describing the development and evaluation of service-oriented architectures and enterprise service bus patterns, how they promote interoperability and discusses scenarios and requirements that lead to successful implementations [22]. A concept design has been outlined by Leventis et al. [23].

The use of software connectors for accessing specific data sources and systems is presented therein. However, for greater homogeneity with the pertinent literature, the authors use here the term "agent" to reflect the actor-based approach in the design and implementation of the proposed solution. There have been many implementations of software agents and agent-oriented systems over the past years with a diverse focus on fields, technology stacks, and features [24,25]. Although this article does not expand into the design of intelligent and autonomous agents, its approach encourages future work where the software agents may be further developed towards such advanced features. To the best of the authors' knowledge, there is no literature that discusses the reuse of existing, open-source editors and tools and their integration to such architecture, especially in the legal and legislative domains, as state-of-the-art projects focus on other research targets such as text classification, named entity recognition, and automated document assembly.

In order to facilitate the editing of legal documents in a standardised scheme, such as Akoma Ntoso, the LEOS authoring tool was selected for two main factors. First, it is based on a promising yet tested usability for editing legal documents utilised or investigated by several public sector institutions in the European Union and beyond. Second, as it is utilising a document repository based on the Content Management Interoperability Services (CMIS) open standard, it provides a vendor-neutral way of storing documents and populating them with pertinent metadata.

The article describes the attempt to repurpose LEOS as an authoring tool to process information located in external sources. By integrating LEOS through customisable agents, its application from law-drafting into the parliamentary control regime is achieved, and thus system interoperability between existing parliamentary control datasets and LEOS. This article presents the use case of integrating LEOS with parliamentary questions of the Hellenic Parliament. The datasets, channelled through the editing software, can be transformed into standardised, and thus open, legislative data. Vice versa, LEOS can also be used as a generic parliamentary control tool that exports XML-based structural elements equivalent to the originally provided data formats. This bi-directional behaviour of the editor, enhanced with supporting agents having the ability to consume heterogeneous data and produce standardised output, contributes to the creation of homogenous linked datasets. Methodologically, an iterative design model was used for the development of the proposed architecture. Frequent exchange with the LEOS development team ensured that the project was in-line with the tool's general specifications.

The LEOS application, its structure and AKN integration, will be shown next, followed by a detailed description of the use case, the handling of written parliamentary questions in the Hellenic Parliament. The proposed platform architecture and the analytic solution design constitute the core of the article, which concludes with a pragmatic evaluation of the presented approach (that is if LEOS, and under which circumstances, is suitable as an authoring tool for exercising parliamentary control). Moreover, the interoperability feature of the tool is assessed.

2. Extending the LEOS System

2.1. The LEOS Tool

The European Commission (EC) has created the ISA^2 Programme (2016–2020) to support the development of IT solutions that bridge or eliminate border and sector differences in public services. In this context, interoperability is a critical parameter [26]. Therefore, under the umbrella of ISA^2, the new European Interoperability Framework (EIF) was created in 2017 as an instrument to provide specific guidance on how to set up interoperable digital public services thus connecting public administrations, businesses and citizens [27]. The ISA^2 programme followed the implementation of ISA (2010–2015); currently, both ISA^2 and the EIF are being evaluated under a strengthened interoperability strategy as identified by the Council of the European Union in its June 2019 Conclusions [10102/19]. LEOS is an ISA^2 solution. Nevertheless, it is not the only existing legal editing system in operation, as displayed in a recent review that analyses legal information systems that support legislative drafting [28].

The EC developed and is currently using a closed-source Microsoft Word addon called LegisWrite, which provides normative wording for fixed structural elements of the legal acts in all 24 official EU languages. As part of the EU Commission's digital interoperability efforts and specifically the Legislation Interoperability Tools action, Legislation Editing Open Software (LEOS) was introduced. This is effectively a government-to-government document management system which provides a web application interface that enables legislation writers to use document templates which predetermined formatting and editable sections, while also introducing an internal versioning system, access controls and an annotation system for tracking author comments during the drafting process. The EC is currently pilot-testing a LEOS instance named "EdiT" to replace LegisWrite.

LEOS is an open-source software available under the European Union Public License (EUPL) that offers the possibility of wide configuration, which makes it an interesting authoring tool for public administrations. Its plain and intuitive WYSIWYG graphical user interface (GUI) takes a lot of formatting burden off the user, focusing the user's involvement away from format to the purely legal part. Tool design makes it possible to design predefined AKN-compatible templates for each category of legal document, such as draft laws, draft presidential decrees, draft ministerial decisions and, in the context of this article, parliamentary questions [29]. The web editor allows users to edit document details such as articles, paragraphs or single passages, while ensuring that content follows specified rules relating to numbering and document structure, for example. Moreover, LEOS supports multiple languages and enables collaborative editing, the use of rich text format (RTF), version control, and input from external sources [30]. The latter is a feature that has caught the authors' attention and has been one of the primary reasons for this investigation.

From the architectural point of view, the system is designed in three distinct layers: the web layer, the service layer, and the backend layer. The web layer (frontend) contains the GUI. The connection with external systems and services such as notification systems, document import, and document conversion is performed at the service level. The backend contains the data repository (database) and content management services [9]. The document repository is based on the Content Management Interoperability Services (CMIS) open standard. As its configuration system is available, it was deemed appropriate to investigate the possibility of extending it with additional document types, as well as determining the feasibility of establishing a content transformation pipeline that would enable the transformation of an unstructured dataset to a structured and easily editable document.

In the legislative ecosystem there are government specific portals, proprietary editors, specific-purpose editors for handling AKN markup, or systems focusing on RDF open linked data and other semantic web technologies. A major detriment to using diverse editors and technologies includes additional training, maintenance overhead and the lack of immediate access. There is a general lack of open source and easily deployable systems that, among others:

- Are focused on legislative document management;
- Can consolidate different workflows in an integrated authoring tool;
- Can be operated by non-technical experts;
- Work with different types of legal documents.

With the appropriate extensions, LEOS could potentially provide a good opportunity to fill these gaps. This is something that is not readily achievable via MS Word plugins or other proprietary systems. Gostojić and Marković [31] provide a recent review of formats, processes, and components that support legal document management, as well as presenting use cases and best-practice suggestions. Their review concludes with recommendations for adopting standardised document formats and publishing generated documents and metadata according to linked open data recommendations. Those recommendations would be covered by the use of an accessible open-source tool that takes advantage of a versatile document format and easily exports content.

2.2. LEOS-AKN Integration

AKN is a popular XML data format for modelling legal parliamentary, legislative, and judicial documents allowing legal data and metadata to be displayed in a fully machine-readable form [32,33]. Now an OASIS standard, it provides for structural (characterisation of parts of the document) and semantic (understanding of the meaning) document markup. In AKN, the legal document can be broadly defined as any text that contains data or information in a structured way. As a result, AKN can have a wide range of applications and can describe texts that record a legal process, such as a bill or a law, meetings of parliamentary (or non-parliamentary) committees, documents granting rights, such as bills, court decisions, memoranda of understanding, parliamentary proceedings, and administrative decisions [34].

As every legal order has a different tradition that diversifies the structure and content of legal documents, there are practically unlimited variations possible. AKN has a proven record of application in different legal orders at the national [35,36] and international level [37,38]. AKN uses the Functional Requirements for Bibliographic Records (FRBR) standard to identify legal texts as bibliographic records. Metadata in AKN are separated from the regulatory part (section meta) to ensure long-term preservation of digital information with legal value and to avoid any misinterpretation due to additional elements. Additionally, AKN can be used together with other vocabularies, such as the European Legislation Identifier [39] or the Data Catalogue Vocabulary [40], to form hybrid ontologies that describe complex systems.

To store the required document templates, configuration metadata and work-in-progress drafts, LEOS uses a document management solution based on Content Management Interoperability Services (CMIS), an OASIS standard enabling information sharing between different content management systems. Integrators may use web services or AtomPub bindings to interact with the data store and expand the domain model as required. Current versions provide for an OpenCMIS InMemory Repository via the H2 database engine and an ISA2 Open CMIS Repo implementation requiring an Oracle database. The content is organised into a templates folder (which contains all of the required structures and configurations to produce new documents) and packages (which are individual collections of documents that are being edited, versioned, and commented upon).

The backbone of the system is a catalog XML, which contains a multilingual, hierarchical description of folders and reusable document structures. An AKN document, containing any required content and metadata, can be loaded in the system and used as a document template. Such a document may contain relationships with other documents that are defined via components and documentRef links. A legislative proposal, for instance, may be linked to an explanatory memorandum. Each of these template files can have a configuration file in JSON format, which provides contextual user guidance for specific elements, or enforce the selection of predetermined alternatives. A structure configuration file can be associated to one or more templates, which may have a similar structure but different content. To enforce a specific structure for each document and limit its editing accordingly, a structure configuration XML is used to describe all elements that can be used in a concrete template, as well as possible interactions between those elements. These elements have a common set of properties to configure their behaviour inside a particular document. Thus, a template can indicate which sections can be deleted or are non-editable (such as a cover page), whether they can be dragged and reorganised, permitted numbering types and many more.

Editing documents on the web layer is conducted using the CkEditor library. The core system includes transformer logic that renders AKN content to HTML. This two-way integration provides the opportunity of easily editing sections via the web interface and saving them into AKN documents that can then be exported either as PDF or archived as LEOS legislation files, such as .zip and .leg containers with additional metadata files. Those .leg files are used as a self-contained snapshot that can be shared between participating institutions and as a reference during the consultation and adoption phases of EU's ordinary legislative procedure.

3. The Parliamentary Questions Use Case

Despite well-documented issues and limitations, parliamentary control belongs to the core functions of a sovereign parliament [41,42]. The parliament exerts control over the executive through special procedures, also called parliamentary control means [43,44]. Written questions are the most frequent means, or instrument, of parliamentary control, something that is also confirmed in the case of the Hellenic Parliament [45]. The latter constitutes the principal field of study of the Hellenic OCR Team, which demonstrated a crowdsourcing methodology for the digital transformation of written questions into open data [14]. Parliamentary questions are legal documents that have a simple, repetitive and defined structure. Consequently, their core elements can be mapped using AKN tags. In the case of the Hellenic Parliament, the core elements of a written question are: sender (MP or MPs); party affiliation; recipient (Minister(s) or Ministry(ies)); date; protocol number(s); subject; and body text, which can be supplemented by logos and/or contact data.

In the context of this article, we evaluate the repurposing of LEOS as an authoring tool for parliamentary control to support written questions from the Hellenic Parliament, initially described as metadata and then reconstructed and managed as exportable LEOS documents. The available data source contains a validated set of written question data in a delimited CSV form, which can be transformed into XML syntax. Every question has a question ID tag, a date and a protocol number. A single question can therefore be asked by multiple MPs. Also, it can be addressed to multiple Ministers or Ministries, with these two elements being redundant. The actual question text is contained in the body text tag, and the filename is kept for document management purposes. UTF-8-character encoding is used to handle the Greek text.

The XML fragment in Appendix A shows an indicative XML representation of a model question. In this model, a 1-to-1 sender to recipient relation is depicted. In real-world parliamentary conditions, parliamentary questions may have many-to-many relationships. The depth of a question tree is not anticipated to exceed two or three levels, depending on whether grouping of questions would be used as a feature or not, for instance if a question is directed towards multiple recipients. The reasoning behind this is that followup questions would be able to be linked with the original question, yet still treated as separate entities. In addition, Appendix B showcases a valid XML markup that is parsable by LEOS and can be loaded as a leos:xml CMIS document, in order to be used as a template for authoring purposes. The Hellenic Parliament logo image that corresponds to this particular implementation and is visible in Figure 3 has been removed for the sake of simplicity. In order for the template to be usable, the catalogue xml needs to be updated and replaced, a paired structure configuration is required to enumerate the table of contents and the element setup, and appropriate CSS style changes need to be added to the codebase.

4. Integration Platform

4.1. Architectural Overview and Solution Design

The proposed architecture is that of an agent-oriented integration platform to encourage and facilitate interoperability at its core. The design aims to enable systems to exchange information and utilise features with each other. This distributed system is realised through specialised software agents for each of the subsystems serving as their interface to the rest of the systems. Software agents are proposed to cover a set of challenges when required to integrate with multiple systems and tackle the volume, location, and diverse format of their available data. They form an in-between layer to canonicalize each system's unique characteristics. This approach aims to address and encapsulate all such heterogeneity based on the logic of the facade pattern that also includes common reusable functionality such as logging, messaging, scheduling, polling mechanisms, communication protocols, throttling, filtering, and data-source types.

Each agent has built in components designed to tackle specific operations which are depicted in Figure 1. A listener component receives messages from other agents through a configurable endpoint and stores the message to its designated queue. A message processor pulls from the queue and processes the message through the appropriate operations depending on the message received. These operations utilise the agent's adapter which directly integrates with the system based on the logic of the adapter pattern. Processing may include features such as delivery guarantees and message prioritization, depending on the specific cases and purposes of the integration. For the purpose of our design, a simple first-come-first-served (FCFS) approach would suffice. Once the respective operations are complete and/or if additional information is required to be received or relayed, any required messages are sent to other agents via the message dispatcher component. Table 1 maps the mentioned patterns to the literature that contains their definition.

Thus, practically any existing systems can be equipped with their own specialised software agent and become qualified to connect with respective agents of other heterogeneous systems. Such systems that would fall into this category could be due to:

- Not having an API of their own;
- Not featuring a push notification feature;
- Pushing change notifications through specific communication protocols;
- Having other unique characteristics that need to be taken into account;
- Having a high volume and variety of data available.

Composite agents may be implemented as orchestrators or aggregators of other agents, forming workflows, automations, aggregation, filtering and throttling. Using composite agents could also work for when a group of agents are preferred to tackle a complex system per subsystem or area of functionality. Their message processors might also encapsulate one-off or recurring behaviours that could include self-triggered sending of messages to other agents to initiate a business process. This approach encourages the reuse of agents implemented by other developers without the necessity to re-engineer them to serve for other use cases. This is achieved through asynchronous flow, throttling of the data transmitted, and filtering based on data relevancy to the specific task at hand. The topology of the agents can be configured by considering factors such as overall performance, location of the data due to privacy issues, time sensitive availability of data due to its volume or expiration, and even migration scenarios. This has a compounding effect in the data flow, specifically in the challenges presented by data volume and diversity and the available data sources and metadata definitions.

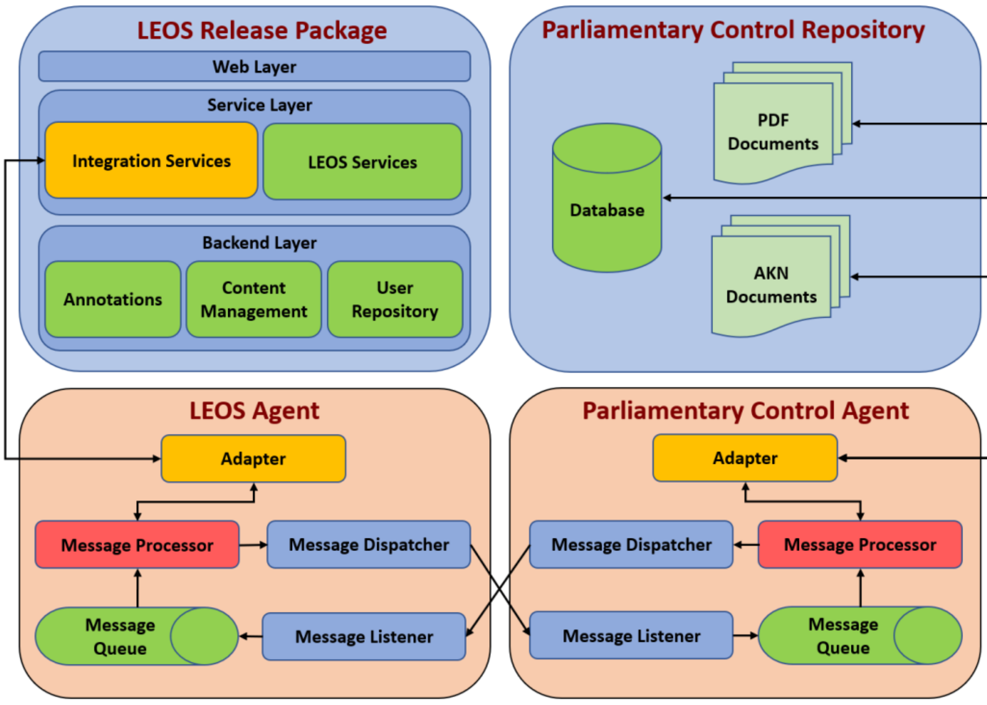


Figure 1. End-to-end conceptual design of the solution.

Table 1. Patterns and their definition.

Pattern	Source
Message Endpoint	[46]
Message Channel	[46]
Request-Reply	[46]
Adapter	[47]
Facade	[47]

In this article, two agents are proposed for connecting the LEOS editor with a validated parliamentary control repository. The Parliamentary Control Agent will be triggered to retrieve parliamentary control data through its adapter component and send them via its message dispatcher to the LEOS Agent. This process could be initiated either from a predefined internal recurring scheduled job within its message processor component or by receiving the respective message to do so by an external system. The LEOS Agent will receive the message containing the data and will then interact with the LEOS editor through its unique adapter component. The adapter will perform the actual operations that will enable LEOS to recognise the documents containing the data. At this point the LEOS editor can be used to edit the documents. Once editing is completed the flow of information could propagate back to the originating data repository, which in this case is the Parliamentary Control repository, following a similar flow. Future work may integrate a usable webapp interface to visually design flows and establish workflows by reusing defined agents, accessing CMIS data and redirecting them to the LEOS-UI for further processing.

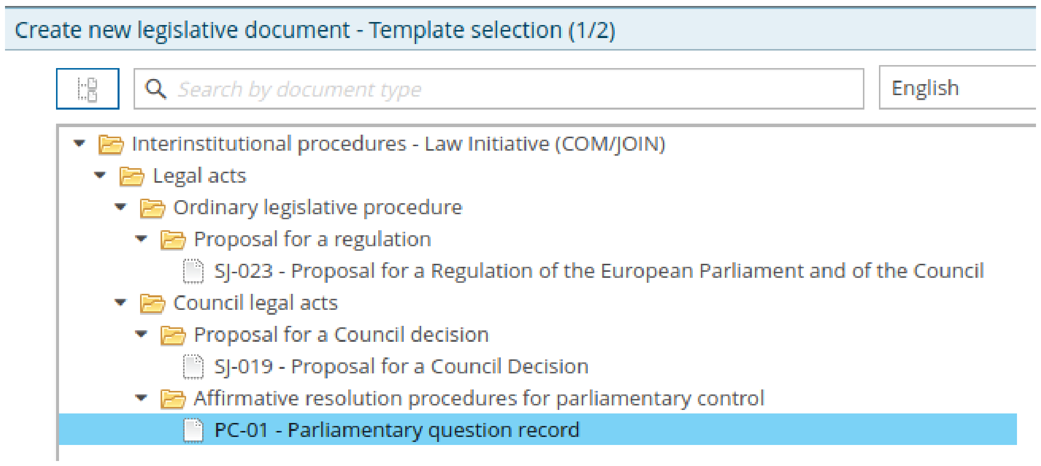
4.2. Adaptors and Technology Stack

Based on the architectural approach, agent implementations should maintain a language and database-agnostic API for agent intercommunication. The attempt described here focused on establishing a proof-of-concept prototype and verify its serviceability as a blueprint for an accessible, reusable and extensible architecture. The initial server-side implementation was based on Python in order to benefit from its package ecosystem and ease of use, and used MongoDB as a document store used as an intermediary storage of transformed data. This approach facilitated rapid development and experimentation, and seems to be compatible with alternative approaches in this area, such as RAWE [48] and LIME [49].

As already mentioned, within the adapter components lie the unique implementation characteristics of the systems being adapted. Two such adapters are required for our use case, for the LEOS editor and the parliamentary control repository. In the case of the parliamentary control repository this would consist of database queries and file operations that will be executed to retrieve the requested data. The implementation will aggregate the information stored in the documents as well as the database in order to produce the results in the required format for the message processor to handle accordingly. In the case of LEOS, appropriate configurations and template definitions are introduced that enhance its web layer with additional document blueprints and serve as prerequisites for the adapter implementation to work.

Any CMIS compatible system can be used to programmatically (or manually) update the document repository. It would also be feasible to automate the initial document creation by utilising said templates for unstructured data provided from external integrations. By using a plethora of CMIS client libraries, a third party may manipulate the data stores to upload any required metadata, document templates and structure configurations, so that a user of the LEOS web interface may then proceed with the drafting process. The default installation lacks an accessible API to automatically introduce new documents or proposal packages via CMIS, but extensions to its framework could expose its internal functionality to do so. Currently, LEOS itself provides a minimal, authenticated API that permits the searching and exporting/rendering of legislation relevant files into PDF or LegisWrite. By reusing existing structures and formatting rules of the LEOS system, the authors were able to generate a proof of concept by introducing the required updates to the XML catalogue and introducing a sample document template.

Figure 2 shows the dialog box 'Create new legislative document—Template Selection' in the editor environment, through which the user may select a template for a new legal document; in this case, a parliamentary question. Templates in LEOS are presented in a tree format. The same step is used to provide the new document with a set of predefined and new metadata, such as the document's title.


Create new legislative document - Template selection (1/2)

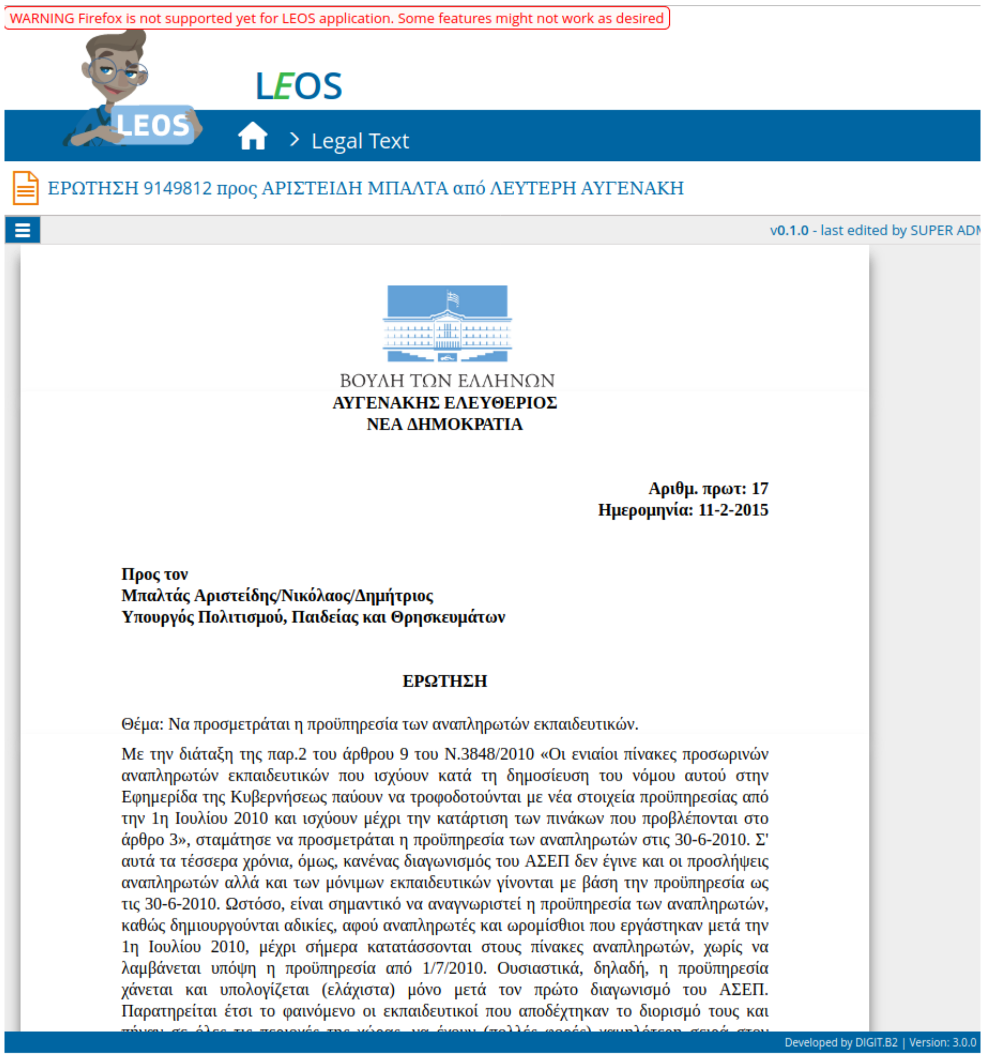
Search by document type | English

- Interinstitutional procedures - Law Initiative (COM/JOIN)
  - Legal acts
    - Ordinary legislative procedure
      - Proposal for a regulation
        - SJ-023 - Proposal for a Regulation of the European Parliament and of the Council
    - Council legal acts
      - Proposal for a Council decision
        - SJ-019 - Proposal for a Council Decision
      - Affirmative resolution procedures for parliamentary control
        - PC-01 - Parliamentary question record

Figure 2. Document wizard for a parliamentary control entry.

Having selected a sample document template and created a new document (a question, ερώτηση [erótisi] in Greek), it can be accessed in edit mode through the question card and the question viewer. Figure 3 shows the result for this proof of concept. Based on a random sample question the newly created document could be successfully populated with information from the data source. For this particular example, the following parliamentary question has been randomly chosen:

question ID = 9149812, submission date = 11 February 2015 and protocol number = 17.


WARNING Firefox is not supported yet for LEOS application. Some features might not work as desired

LEOS

Legal Text

ΕΡΩΤΗΣΗ 9149812 προς ΑΡΙΣΤΕΙΔΗ ΜΠΑΛΤΑ από ΛΕΥΤΕΡΗ ΑΥΓΕΝΑΚΗ

v0.1.0 - last edited by SUPER ADI

ΒΟΥΛΗ ΤΩΝ ΕΛΛΗΝΩΝ
ΑΥΓΕΝΑΚΗΣ ΕΛΕΥΘΕΡΙΟΣ
ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ

Αριθμ. πρωτ: 17
Ημερομηνία: 11-2-2015

Προς τον
Μπαλτάς Αριστείδης/Νικόλαος/Δημήτριος
Υπουργός Πολιτισμού, Παιδείας και Θρησκευμάτων

ΕΡΩΤΗΣΗ

Θέμα: Να προσμετράται η προϋπηρεσία των αναπληρωτών εκπαιδευτικών.

Με την διάταξη της παρ.2 του άρθρου 9 του Ν.3848/2010 «Οι ενιαίοι πίνακες προσωρινών αναπληρωτών εκπαιδευτικών που ισχύουν κατά τη δημοσίευση του νόμου αυτού στην Εφημερίδα της Κυβερνήσεως παύουν να τροφοδοτούνται με νέα στοιχεία προϋπηρεσίας από την 1η Ιουλίου 2010 και ισχύουν μέχρι την κατάρτιση των πινάκων που προβλέπονται στο άρθρο 3», σταμάτησε να προσμετράται η προϋπηρεσία των αναπληρωτών στις 30-6-2010. Σ' αυτά τα τέσσερα χρόνια, όμως, κανένας διαγωνισμός του ΑΣΕΠ δεν έγινε και οι προσλήψεις αναπληρωτών αλλά και των μόνιμων εκπαιδευτικών γίνονται με βάση την προϋπηρεσία ως τις 30-6-2010. Ωστόσο, είναι σημαντικό να αναγνωριστεί η προϋπηρεσία των αναπληρωτών, καθώς δημιουργούνται αδικίες, αφού αναπληρωτές και ωρομίσθιοι που εργάστηκαν μετά την 1η Ιουλίου 2010, μέχρι σήμερα κατατάσσονται στους πίνακες αναπληρωτών, χωρίς να λαμβάνεται υπόψη η προϋπηρεσία από 1/7/2010. Ουσιαστικά, δηλαδή, η προϋπηρεσία χάνεται και υπολογίζεται (ελάχιστα) μόνο μετά τον πρώτο διαγωνισμό του ΑΣΕΠ. Παρατηρείται έτσι το φαινόμενο οι εκπαιδευτικοί που αποδέχτηκαν το διορισμό τους και

Developed by DIGIT.B2 | Version: 3.0.0

Figure 3. Web UI rendition based on the uploaded document.

It belongs to the 16th parliamentary period of the Hellenic Republic and can be accessed through the portal of the Hellenic Parliament.

5. Evaluation

LEOS represents a versatile legal informatics solution that uses AKN to transform legal texts into BOLD. The article presented ongoing research to repurpose the tool to a parliamentary control authoring solution that generates written parliamentary questions of a predefined structure according to a standard template. In terms of the default functionality, such as making bill proposals or adding annexes to existing documents, LEOS is accurately providing constrained document templates for use by EU entities, such as EU member state parliaments. Once the above knowledge was acquired, the system's data definitions and architecture permitted the authors to define new templates to support parliamentary questions and proceed with the generation of respective documents. It is now feasible for any CMIS compatible mechanism to produce and consume these legal documents. Together with the ability to define and add appropriate metadata, the architecture can support a scalable way towards storing and processing BOLD.

The interoperability dimension of LEOS was evaluated using an enterprise integration patterns approach. Specialised software agents with reusable components were proposed to connect the LEOS system with other external data sources. By having dedicated agents per system, including LEOS, additional supported systems could be introduced without requiring re-implementation or adjustments of the rest of the agents. More importantly, load balancing, throttling, and asynchronous operations can be applied within those agents to handle the volume of data available. This includes the ability of having multiple instances of LEOS connecting to a single external system, so that a group of users can concurrently view, process, and save the relevant data back to that system. Hence, the architecture can be configured and extended from any junction in the data flow to harness big data.

The Hellenic Parliament control data formed the use case for this study. Architectural and design elements were presented to serve as a basis for an end-to-end implementation that may perform the complete integration pipeline of data to the repository and back again after being processed within the LEOS editor. The end result of the current development is a minimal implementation of a parliamentary question document template to facilitate the elaboration of the proposed approach, without additional structure or guidance configurations for the editor. The CMIS updates were successful, as well as the export functionality. Potential future work may introduce document types and business logic in LEOS to natively support the rendering and editing of debate AKN documents, as well as the creation of a document collection that could additionally include an answer document from the questioned MP to be paired with the relevant question.

Initial data clean-up and transformation into the required metadata structure provided for insights into the designs of the agents and storing of data in the MongoDB backend. Nevertheless, the initial learning curve of LEOS' internals proved steeper than expected, thus extending development time significantly and in effect reduced the initial scope of an end-to-end implementation. Rapid iteration and experimentation ware impaired by the fact that the currently released LEOS version does not perform hot reloads of data and configurations, but only loads them during initialisation. Thus, configuration changes and newly added documents could not be parsed and viewed on the web layer without restarting the relevant services. This was an unexpected challenge and, once solved, would constitute a major enhancement in the system's usability. Similarly, the default package provides an InMemory H2 database without file persistence, which would require manual application of changes on the file system in order to be available on the next initialisation. Thus, the quickest setup requires installation and configuration of an Oracle Database Express Edition instance. Potential improvements would be to integrate an H2 component with automatic file persistence or providing easier integration with additional and more accessible database backends, such as SQLite.

In its current state, the proposed solution does not consider any authentication or access controls schemes. The suggested architecture would support any additional security or authentication safeguards. The current version of LEOS provides a security assertion markup language (SAML) component that can be integrated with existing identity providers. It also provides roles for permission management, which can limit document editing, downloads and visibility.

From prototype to its current 3.0.0 version, LEOS has grown in complexity and has received significant contributions from stakeholders, now constituting a community of users and supporters [50]. Nevertheless, there are currently three concerns for early adoption. The first is that a major part of business logic is hardcoded into its codebase, requiring software engineering work on the Spring framework to modify. Secondly, it is using Vaadin v8 as its web framework that would require extensive work to modernise to its current Long-Term Support version 14 (or newer). Finally, and most importantly, future development milestones will reengineer the system to use a modern angular frontend, will externalise more of its business logic into configuration files, provide RESTful service endpoints, and will expose individual document controllers, which would greatly enhance the usability of the editor and facilitate further integrations.

6. Conclusions

Following the architecture and design of the integration platform presented in this article along with the required actions to satisfy prerequisites, such as LEOS configuration and templates, an end-to-end implementation is the major subsequent project that needs to be worked on. Although the design provides for an expandable implementation and does not limit the user to a specific technology stack due to its distributed and message-oriented approach, it also adds to the complexity of introducing additional components to the overall system. Hence, certain features also need to be implemented as part of the overall framework to mitigate the added complexity. These potential drawbacks could be mitigated by providing common implementations that may be utilised to cover required or frequent use scenarios.

The implementation should include a set of reusable components that will pave the way for a reduced development effort of additional agents. These additional agents should be implemented based on quickstart templates that consist of built-in functionality for specific types of agents, such as those interacting with a database or file system or even other integration platforms such as Zapier, ITTT, and MuleSoft as well as other agent-oriented systems and bots, such as chatbots. Agents whose implementation is open-source can be cloned and used as a starting point for other agent implementations. Open sourcing the Parliamentary Control Agent is intended once implementation is complete, thus eventually contributing to the LEOS codebase to support the working process of parliamentary questions in the European Parliament. Future work should thus enable and encourage an ecosystem (and potentially a marketplace) of interested parties to add additional agents that can integrate with existing ones.

Author Contributions: All authors conceived, designed, and performed the experiments, analyzed the data, and wrote the article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data source used in this article is openly accessible in Zenodo at doi:10.5281/zenodo.4747451.

Acknowledgments: The authors would like to thank the Hellenic OCR Team members who compiled and validated the dataset of parliamentary questions that was used to test the research hypothesis. The authors very much appreciate the support by the LEOS development team, which provided useful comments and insights towards the final drafting phase of the article.

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

Appendix A

The code fragment below presents the representative XML syntax of the mentioned data set [14].

```
<data>
        <field_1>question ID</field_1>
        <field_2>type of control: Question</field_2>
        <field_3>submission date</field_3>
        <field_4>protocol number</field_4>
        <field_5>subject</field_5>
        <field_6>
             <Item>
                     <ID>MP ID</ID>
                     <Name>name of MP</Name>
             </Item>
        </field_6>
        <field_7>
             <Item>
                     <Name>name of Ministry</Name>
             </Item>
        </field_7>
        <field_8>
             <Item>
                     <Name>questioned Minister (Ministry name) </Name>
             </Item>
        </field_8>
        <field_9>
             <Item>
                     <Name>parliamentary group</Name>
             </Item>
        </field_9>
        <field_10>
             <Item>
                     <ID>file_name_ID</ID>
                     <File>body text</File>
             </Item>
        </field_10>
</data>
```

Appendix B

This code fragment shows an AKN structure representation sample for an editable parliamentary question.

```
<?xml version="1.0" encoding="UTF-8"?>
                <akomaNtoso xmlns="http://docs.oasis-open.org/legaldocml/ns/akn/3.0"
                    xmlns:leos="urn:eu:europa:ec:leos"
                    xmlns:xml="http://www.w3.org/XML/1998/namespace">
                    <debate name="ParliamentaryQuestion">
                        <meta>
                           ...
                        </meta>
                        <preface xml:id="preface">
                            <container name="logo" xml:id="coverpage__container_1">
                                <img xml:id="coverpage__container_1__p__img" src="#hellenic_parlia-
        ment_logo"/>
                            </container>
                            <container name="docLabel" xml:id="preface__container_1">
                                <p xml:id="preface__container_1__p">
                            ΑΥΓΕΝΑΚΗΣ ΕΛΕΥΘΕΡΙΟΣ <br/>
                        </p>
                        <p xml:id="preface__container_2__p">ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ</p>
                    </container>
                    <container>
                        <p xml:id="protocol_number">Αριθμ. πρωτ: ...</p>
                        <p xml:id="submission_date">Ημερομηνία: ...</p>
                    </container>
                    <container>
                        <p xml:id="question_to">Προς τ...</p>
                        <p xml:id="question_to_mp">...</p>
                        <p xml:id="question_to_mp_ministry">...</p>
                    </container>
                    <container>
                        <p xml:id="question">ΕΡΩΤΗΣΗ</p>
                    </container>
                    <p xml:id="question_subject" style="text-align: left;">Θέμα: </p>
                </preface>
                <debatebody>
                    <questions xml:id="debate__debateBody__questions">
                        <debatesection xml:id="debate__debateBody__questions__debateSection">
                            <question by="~questioning_mp_name" xml:id="debate__debateBody__ques-
tions__debateSection__question">
                                <blockcontainer xml:id="debate__debateBody__questions__debateSec-
tion__question__blockContainer">
                                    <block xml:id="debate__debateBody__questions__debateSec-
tion__question__blockContainer__block">
        <p name="question_text" xml:id="debate__debateBody__questions__debateSection__ques-
tion__blockContainer__block__p_1">
        ...
        </p>
                                    </block>
                                </blockcontainer>
                            </question>
                        </debatesection>
                    </questions>
                </debatebody>
                <conclusions xml:id="conclusions">
                    <block name="signatory" xml:id="conclusions__block_1">
                        <signature xml:id="conclusions__block_1__signature_1">
                            <person xml:id="conclusions__block_1__signature_1__person" re-
fersTo="~questioning_mp_name">...</person>
        <organization xml:id="conclusions__block_1__signature_1__organization" refersTo="~ques-
tioning_mp_group">...</organization>
                        </signature>
                    </block>
                </conclusions>
            </debate>
        </akomaNtoso>
```

References

1. Dawes, S.; Cresswell, A.; Pardo, T. From "need to know" to "need to share": Tangled problems, information boundaries, and the building of public sector knowledge networks. *Public Adm. Rev.* **2009**, *69*, 392–402. [CrossRef]
2. Sartor, G.; Palmirani, M.; Francesconi, E.; Biasiotti, M.A. *Legislative XML for the Semantic Web: Principles, Models, Standards for Document Management*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011; Volume 4. [CrossRef]

3. Antoniou, G.; Van Harmelen, F. *A Semantic Web Primer*; MIT Press: Cambridge, MA, USA, 2004; Available online: https://mitpress.mit.edu/books/semantic-web-primer (accessed on 17 September 2021).
4. Francoli, M. Parliaments Online: Modernizing and Engaging? In *Workshop Parliaments in the Digital Age at the Oxford Internet*; Oxford Internet Institute (OII): Oxford, UK, 2008; Volume 13, pp. 4–10. Available online: https://www.oii.ox.ac.uk/archive/downloads/publications/FD13.pdf (accessed on 17 September 2021).
5. Biasiotti, M.; Francesconi, E.; Palmirani, M.; Sartor, G.; Vitali, F. *Legal Informatics and Management of Legislative Documents*; Global Centre for ICT in Parliament: Rome, Italy, 2008; Volume 2, Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.183.2734&rep=rep1&type=pdf (accessed on 17 September 2021).
6. Akoma Ntoso. XML for Parliamentary, Legislative & Judiciary Documents. Available online: http://www.akomantoso.org/ (accessed on 17 September 2021).
7. Akn4eu—EU Vocabularies—Publications Office of the EU. Available online: https://op.europa.eu/en/web/eu-vocabularies/dataset/-/resource?uri=http://publications.europa.eu/resource/dataset/akn4eu (accessed on 17 September 2021).
8. AT4AM.Eu. Available online: https://at4am.eu/ (accessed on 17 September 2021).
9. LEOS—Open Source Software for Editing Legislation | Joinup. Available online: https://joinup.ec.europa.eu/collection/justice-law-and-security/solution/leos-open-source-software-editing-legislation (accessed on 17 September 2021).
10. Malanga, K.; Mehat, J.; Ganchev, I.; Wandeto, J.; Shikali, C. Evaluation of Open Source Software with QualiPSO OMM: A case for Bungeni and AT4AM for All. In Proceedings of the FOSCC-15, Muscat, Oman, 18–19 February 2015; Available online: http://41.89.227.156:8080/xmlui/handle/123456789/256 (accessed on 17 September 2021).
11. Fitsilis, F. Structuring a Course on Legal Interoperability | Joinup. 3 August 2020. Available online: https://joinup.ec.europa.eu/collection/digital-skills-public-sector/news/hellenic-case-report (accessed on 17 September 2021).
12. HellenicOCRteam | A Crowdsourcing Initiative. Available online: https://hellenicocrteam.gr/ (accessed on 17 September 2021).
13. Fitsilis, F.; Koryzis, D.; Svolopoulos, V.; Spiliotopoulos, D. Implementing digital parliament innovative concepts for citizens and policy makers. In *HCI in Business, Government and Organizations. Interacting with Information Systems (HCIBGO)*; Nah, F.H., Tan, C.H., Eds.; Springer: Cham, Switzerland, 2017; pp. 154–170. [CrossRef]
14. Fitsilis, F.; Saalfeld, T.; Schwemmer, C. Content Reconstruction of Parliamentary Questions. Combining Metadata with an OCR Process. In Proceedings of the 5th International Virtual Conference on Advanced Scientific Results, Virtual, 26–30 June 2017; Volume 5, pp. 107–112. Available online: https://career.duth.gr/portal/?q=node/39414#.YUVfnn0RWUl (accessed on 18 September 2021). [CrossRef]
15. Winkels, R. The openlaws project: Big open legal data. In Proceedings of the 18th International Legal Informatics Symposium, Salzburg, Austria, 26–28 February 2015; pp. 189–196. Available online: https://hdl.handle.net/11245/1.483455 (accessed on 17 September 2021).
16. Stavropoulou, S.; Romas, I.; Tsekeridou, S.; Loutsaris, M.; Lampoltshammer, T.; Thurnay, L.; Virkar, S.; Schefbeck, G.; Kyriakou, N.; Lachana, Z.; et al. Architecting an innovative big open legal data analytics, search and retrieval platform. In Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance, Athens, Greece, 23–25 September 2020; pp. 723–730. [CrossRef]
17. Cuquet, M.; Fensel, A. The societal impact of big data: A research roadmap for Europe. *Technol. Soc.* **2018**, *54*, 74–86. [CrossRef]
18. Duan, Y.; Edwards, J.S.; Dwivedi, Y.K. Artificial Intelligence for Decision Making in the Era of Big Data—Evolution, Challenges and Research Agenda. *Int. J. Inf. Manag.* **2019**, *48*, 63–71. [CrossRef]
19. Devins, C.; Felin, T.; Kauffman, S.; Koppl, R. The law and big data. *Cornell JL Public Policy* **2017**, *27*, 357–413.
20. Zimmermann, O.; Pautasso, C.; Hohpe, G.; Woolf, B. A decade of enterprise integration patterns: A conversation with the authors. *IEEE Softw.* **2015**, *33*, 13–19. [CrossRef]
21. Petrasch, R. Model-based engineering for microservice architectures using Enterprise Integration Patterns for inter-service communication. In Proceedings of the 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhon Si Thammarat, Thailand, 12–14 July 2017; pp. 1–4. [CrossRef]
22. Kurth, W. *Evaluating the Operational, Performance, and Extensibility Characteristics of SOA, ESB, and Microservices Architectures*; Capstone Report; University of Oregon: Eugene, OR, USA, 2017; Available online: https://scholarsbank.uoregon.edu/xmlui/bitstream/handle/1794/23233/Kurth_2017.pdf (accessed on 18 September 2021).
23. Leventis, S.; Anastasiou, V.; Fitsilis, F. Application of Enterprise Integration Patterns for the digital transformation of parliamentary control. In Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance, Athens, Greece, 23–25 September 2020; pp. 738–741. [CrossRef]
24. Cranefield, S.; Ranathunga, S. Embedding Agents in Business Processes Using Enterprise Integration Patterns. In *Engineering Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 97–116. [CrossRef]
25. Pal, C.V.; Leon, F.; Paprzycki, M.; Ganzha, M. A Review of Platforms for the Development of Agent Systems. *arXiv* **2020**, arXiv:2007.08961.
26. Kalogirou, V.; Charalabidis, Y. The European union landscape on interoperability standardisation: Status of European and national interoperability frameworks. In *Enterprise Interoperability VIII. Proceedings of the I-ESA Conferences*; Popplewell, K., Thoben, K.D., Knothe, T., Poler, R., Eds.; Springer: Cham, Switzerland, 2019; Volume 9, pp. 359–368. [CrossRef]
27. Kouroubali, A.; Katehakis, D. The new European interoperability framework as a facilitator of digital transformation for citizen empowerment. *J. Biomed. Inform.* **2019**, *94*, 103166. [CrossRef] [PubMed]

28. Loutsaris, M.; Charalabidis, Y. Legal informatics from the aspect of interoperability: A review of systems, tools and ontologies. In Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance, Athens, Greece, 23–25 September 2020; pp. 731–737. [CrossRef]
29. LEOS Document Templates and Elements Configuration. Available online: https://github.com/l-e-x/leos/blob/master/docs/Document-templates-and-configuration/LEOS_Document_templates_and_elements_configuration_manual.pdf (accessed on 17 September 2021).
30. LEOS User Guide. Available online: https://joinup.ec.europa.eu/sites/default/files/inline-files/Leos%203.0%20Manual_0.pdf (accessed on 17 September 2021).
31. Gostojić, S.; Marković, M. Legal Document Management: An Integrated Approach. In Proceedings of the Sinteza 2019—International Scientific Conference on Information Technology and Data Related Research, Belgrade, Serbia, 20 April 2019; pp. 374–380. [CrossRef]
32. Palmirani, M.; Vitali, F. Akoma-Ntoso for legal documents. In *Legislative XML for the Semantic Web*; Springer: Dordrecht, The Netherlands, 2011; Volume 4, pp. 75–100. [CrossRef]
33. Koniaris, M.; Papastefanatos, G.; Vassiliou, Y. Towards Automatic Structuring and Semantic Indexing of Legal Documents. In Proceedings of the 20th Pan-Hellenic Conference on Informatics, Patras, Greece, 10–12 November 2016; Volume 4, pp. 1–6. [CrossRef]
34. Palmirani, M.; Vitali, F. *Legislative XML: Principles and Technical Tools*; Inter-American Development Bank: Washington, DC, USA, 2012; Available online: https://publications.iadb.org/en/legislative-xml-principles-and-technical-tools (accessed on 17 September 2021).
35. Gen, K.; Akira, N.; Makoto, M.; Yasuhiro, O.; Tomohiro, O.; Katsuhiko, T. Applying the Akoma Ntoso XML schema to Japanese legislation. *J. Law Inf. Sci.* **2015**, *24*, 49–70.
36. Palmirani, M.; Vitali, F.; Bernasconi, A.; Gambazzi, L. *Swiss Federal Publication Workflow with Akoma Ntoso*; IOS Press: Amsterdam, The Netherlands, 2014; pp. 179–184.
37. Peroni, S.; Palmirani, M.; Vitali, F. UNDO: The United Nations System Document Ontology. In *The Semantic Web—ISWC 2017*; Springer: Cham, Switzerland, 2017; pp. 175–183. [CrossRef]
38. Palmirani, M. Akoma Ntoso for Making FAO Resolutions Accessible. In *Knowledge of the Law in the Big Data Age*; Peruginelli, G., Faro, S., Eds.; IOS Press: Amsterdam, The Netherlands, 2019; pp. 159–169. [CrossRef]
39. Eli_xml—European Legislation Identifier—EU Vocabularies—Publications Office of the EU. Available online: https://op.europa.eu/en/web/eu-vocabularies/dataset/-/resource?uri=http://publications.europa.eu/resource/dataset/eli_xml (accessed on 17 September 2021).
40. Data Catalog Vocabulary. 2020. Available online: https://www.w3.org/TR/vocab-dcat-2/ (accessed on 17 September 2021).
41. Saalfeld, T. Members of parliament and governments in Western Europe: Agency relations and problems of oversight. *Eur. J. Political Res.* **2000**, *37*, 353–376. [CrossRef]
42. Strøm, K.; Müller, W.; Smith, D. Parliamentary control of coalition governments. *Annu. Rev. Political Sci.* **2010**, *13*, 517–535. [CrossRef]
43. Yamamoto, H. *Tools for Parliamentary Oversight: A Comparative Study of 88 National Parliaments*; Inter-Parliamentary Union: Geneva, Switzerland, 2007; Available online: http://archive.ipu.org/pdf/publications/oversight08-e.pdf (accessed on 17 September 2021).
44. Pelizzo, R.; Stapenhurst, F. *Parliamentary Oversight Tools: A Comparative Analysis*; Routledge: London, UK, 2011. [CrossRef]
45. Fitsilis, F.; Koryzis, D. Parliamentary control of Governmental actions on the interaction with European organs in the Hellenic Parliament and the National Assembly of Serbia. Online Papers on Parliamentary Democracy. 2016. Available online: https://www.pademia.eu/publications/online-papers-on-parliamentary-democracy/online-papers-on-parliamentary-democracy-v2016/ (accessed on 17 September 2021).
46. Hohpe, G.; Woolf, B. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*; Addison-Wesley Professional: Boston, MA, USA, 2004.
47. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison-Wesley Professional: Boston, MA, USA, 1994.
48. Palmirani, M.; Cervone, L.; Bujor, O.; Chiappetta, M. RAWE: A web editor for rule markup in LegalRuleML. Available online: http://ceur-ws.org/Vol-1004/paper4.pdf (accessed on 17 September 2021).
49. Gheen, T. A New Akoma Ntoso Tool: The LIME Editor | In Custodia Legis: Law Librarians of Congress. 3 April 2014. Available online: https://blogs.loc.gov/law/2014/04/a-new-akoma-ntoso-tool-the-lime-editor/ (accessed on 17 September 2021).
50. Vasilescu, A. User from Spain: LEOS Will Help Us Make the Drafting of Legal Texts More Efficient | Joinup. 29 January 2020. Available online: https://joinup.ec.europa.eu/collection/justice-law-and-security/solution/leos-open-source-software-editing-legislation/news/leos-reuse-story (accessed on 17 September 2021).