Automatic Generation of Web Applications for Information Systems

Yalmar Ponce Atencio  
Escuela Profesional de Ingeniería de Sistemas, Universidad Nacional José María Arguedas  
E-mail: yalmar@unajma.edu.pe

Manuel J. Ibarra  
Escuela Profesional de Ingeniería Informática y Sistemas, Universidad Nacional Micaela Bastidas de Apurímac  
E-mail: manuelli@unap.edu.pe

Julio Huanca Marin  
Escuela Profesional de Ingeniería de Sistemas, Universidad Nacional José María Arguedas  
E-mail: julio@unajma.edu.pe

Edgar Holguín Holguín  
Escuela Profesional de Ingeniería de Sistemas, Universidad Nacional del Altiplano  
E-mail: eholguin@unap.edu.pe

Abstract. The use of automation tools for software development or so-called Computer Aided for Automatic Software Construction tools is a sign of a high level of maturity for the software development process, and is especially necessary when is needed to build information systems, where many parts or modules of a bigger projects have similar features and functionalities. Among others, Automatic Software Construction tools often include code generation software based on templates. Template-based code generation involves the automatic creation of source code or some code output corresponding to a predefined format called a template. In this article, a software architecture for automatic software construction is presented. The tool allows prototyping software modules, projects and functional code to different needs and purposes. The basis of this technical solution is the creation of a general template representation format, which includes many common source code, configuration data files, and a function library that allows you to interpret the input model and use the template for creating objects from different types based on the input data. The generation algorithm and the automatic integration method use a database model in order to define the main behavior of the generated software. The achieved results shown the creation of complete functional web applications, and since the complete source code is generated, is possible customize for improve or add different or specific functionalities.

Keywords: Computer-aided and rapid software development, template-based code generation, model-driven engineering, automatic programming, interpretation of text data.
1. Introduction

As the processing capabilities of computers have grown exponentially as have the possibilities of communication and data transfer, it is essential to develop agile solutions that adapt to the work environment. There are several programming languages that allow developing solutions in an agile way, among which we can highlight: Java, Ruby, PHP, Python, Go, JavaScript, among others. The main characteristics that are taken into account when choosing a programming language are: development productivity, software quality, program portability, libraries and components support, integration components, among others. However, choosing a good programming language does not guarantee a high-quality software product, since development is a task that requires a high technical knowledge of the programming language by the developer team, in order to implement solutions with good results, based on good practices. For most programming languages, there are various frameworks, which are schemes for the development or implementation of applications. Then, Choosing one or several frameworks that facilitate the development of web applications, could be determinant for the success or failure, for this reason it is important to know the advantages and disadvantages of the different frameworks available before chose them [1]. Among the most popular programming languages is Python, which is a fairly simple programming language but at the same time of great potential, and in the field of web application development, there are several frameworks based on this language, being the Django framework the most popular. In recent years, has had a positive reception by web developers and is above all other frameworks based on the Python programming language. In the current market, there are many web-oriented frameworks for many programming languages, which have the functionality to generate code such as: Ruby on Rails, Django, Laravel, Yii2, among others. These frameworks make coding easier for developers. The popularity of frameworks and programming languages is very varied, finding advantages and disadvantages, and in most cases it lies with the skills of developers, where depending on that skills, using a certain language or framework could get better results. However, the vast majority of developers do not have great skills, and developing an application can take considerable time, even using popular and modern programming languages and frameworks. On the other hand, there is a great demand to develop applications that handle information stored in databases, using the Model-View-Controller (MVC) pattern [2] or variants, where the business logic is based on the database, and the user interface is practically the most important thing of the application. In that context, the development process of these applications is similar, depending on the physical model of the database. Based on that similar process, for many applications, it is possible to identify the common aspects shared by this type of applications to create a general template to generate a skeleton of the application, and then be able to customize it according to the particular features of the physical model from the database.

Based on the analyzes of the frameworks mentioned above, a generator is required that allows creating web applications in short periods of time with a structure based on good practices. It is known that most web applications have been developed with the JavaScript programming language, therefore it is very convenient to use frameworks based on it, and on the other hand, we know that most applications that handle data are migrating from the desktop type to the web, and its characteristics should be as similar as possible and with the same functionalities. For this reason, it is important to use a framework that allows creating applications with almost identical user interface to desktop applications. This research details a template-based generative technique that allows the creation of web applications with responsive interfaces, which can be run on any device that has a web browser. The application code is automatically generated by selecting a database and the tables with which you want to interact. The implemented web application generator application incorporates tools to incorporate advanced functionalities such as dependent tables, which is common in relational models. As achieved results, some real applications are presented, namely, a documentary management system, enrollment management
system, registration system and assistance system. Additionally, other types of themes such as bootstrap can be easily incorporated.

The rest of the article is organized as follows: In the section II are presented the related works, in the Section III the proposed methodology is presented; Section IV presents implementation details; section V presents the results and finally in section VI the conclusions and future works.

2. Related works
Currently, having an increasingly demand for building web applications, during the development process, the automation level of software development can be improved by using the efficient methods and the automation techniques. The most important goals of software development are to reduce the cost of software development to increase the efficiency of software generation and to improve the quality of software products. At present, automatic code generation could reduce programmers' work greatly and improve software development efficiency. Therefore, automatic code generation has become focus research. Currently, artificial intelligence methods are being investigated to generate code automatically [3, 4], however, these techniques are in a very initial stages, some of them focusing on natural language processing.

Going back to the traditional, source code completion is a typical application of automatic code generation, which can help programmers write source code. Source code completion has become the main part of modern integrated development environment. The integrated development environments for different programming languages allow the programmer to complete the rest of the program while typing the source code, such as Eclipse, Visual Studio Code and others [5].

Model-Driven or model based approaches aim to improve the Web applications development process by focusing on modeling instead of coding, and generating the running application by transformations from conceptual models or from database physical models to code. The Interaction Flow Modeling Language (IFML) has been an important milestone in the evolution of Web modeling languages, indicating not only the maturity of the field but also a final convergence of languages. Brambilla, et al. [6] have presented and interesting work using IFML for code generation of user interfaces. Another important contribution is presented by Rossi, et al. [7], they explain the evolution of modeling and design approaches since the early years (the 90's) detailing the forces which drove that evolution and discussing the strengths and weaknesses of some of those approaches and obstacles that hinder the dissemination of model-driven techniques in the Web engineering field.

Software development tools have growing impact on the effective and efficient development of software-intensive systems. Modeling can be used throughout the entire system development life-cycle. Modern software development methodologies rely on building conceptual models of the software system and analyzing the models before translation to conventional programming language format. Also, the concept of using Model Based Development has become central to design and verification practices. Supporting the approach software design tools provide additional Automatic Code Generation capability can be treated as the next logical step in specifying and describing a software system in the progression: from machine language to assembly language to algorithmic high-level language to object-oriented high-level language to graphic modeling language [8].

A critical step in model-driven engineering (MDE) is the automatic synthesis of a textual artifact from models. This is a very useful model transformation to generate application code, to serialize the model in persistent storage, and generate documentation or reports. Among the various model-to-text transformation paradigms, template-based code generation (TBCG) is the most popular in MDE [9, 10, 11]. TBCG is a synthesis technique that produces code from high-level specifications, called templates. A template is an abstract and generalized representation of the textual output it describes. It has a static part, text fragments that appear in the output...
“as is”, and a dynamic part embedded with splices of meta-code that encode the generation logic. It is a popular technique in MDE, as they both emphasize abstraction and automation. Given the diversity of tools and approaches, it is necessary to classify and compare existing TBCG techniques and tools to provide appropriate support to developers.

Nowadays is appropriate to conduct a study in order to understand the trends, identify the characteristics of TBCG, assess the popularity of existing tools, and determine the influence that MDE has had on TBCG over the past decades. Based on this study, is possible to compare the most popular TBCG tools. We can perform a qualitative evaluation of their expressiveness based on typical metamodel patterns that influence the implementation of the templates. The expressiveness of a tool is the set of language constructs that can be used to complete a particular task natively. This is important since, to the best of our knowledge, there are no available metrics to assess the code generation templates. Also is possible to evaluate the performance and scalability of these tools based on a range of models that conform to a metamodel composed by the combination of these patterns [12].

Likewise, Web services backed by a database is a mature system architecture. In the context of Mobile Computing, Cloud Computing, Internet-Of-Things, developers need to build more and more web services quickly and correctly. There are vast number of web applications which requires database system to function. A web application frequently needs a database to store the data. Web apps are typically structured into logical tiers. For example, a common structure uses three tiers. The first tier is the web browser, which is responsible for presenting the user interface. The middle tier is an application server also called as business layer, which is responsible for the application’s functionality. The third tier is a database server or file system, which is responsible for data storage. Web services are a breed of Web applications, where performs functions that can be anything from simple requests for information to creating and executing complicated business processes. The vast increasing number of Web services is transforming the Web from a data-oriented repository to a service-oriented repository . In many frameworks, existing business logic will be wrapped as Web services that would be accessible on the Web via a Web service middleware. The lack of well-defined semantics affects their ability to achieve seamless interoperability in the true sense. Hence, when designing a web application, the goals of a software architect are to minimize the complexity while designing a secure, high-performance application using appropriate source code that can be provided by the template based code generator [13].

The fierce competition and budgetary pressures have motivated software companies to distribute their development activities out to the entire world. This type of development in which team members belong to different countries is termed as Global Software Development (GSD). The main motivation behind GSD is the desire to reduce cost of development by utilizing pool of low-salaried, skilled software engineers belonging to less developed economies. Other impetus includes closer proximity to customer, reduced time to market by exploiting time zone differences, improved work modularization, innovation and learning. However, the potential benefits of GSD are only partially achieved due to several distances that interfere with management and execution of these projects. The distances that interplay between distributed teams are geographical, temporal, socio-cultural, and organizational which result into communication, coordination, control, and collaboration challenges representing other kind of difficulties [14].

In software engineering, optimization of the development process and quality assurance are important. The objective of the optimization is to avoid repetitive activities, which cause loss of time and delay in delivery periods. With the increasing use of the internet, web applications must have functionalities similar to those of traditional desktop applications, in order to satisfy the needs of customers, for this reason; Developing applications with rich interfaces that are easy to use and scale is important. In this research, a fully functional application generator is presented, based on a simple, intuitive and user-friendly interface. The generator is based on the
ExtJs Framework, based on the JavaScript programming language and the MySQL database management system is used for the information processing. The results of the application generator demonstrate fully functional web applications can be generated in seconds, with complete user interaction, secure and with professional look.

3. Methodology
The used methodology in this research work comprises three stages. On the first stage, a review of research works related to Automatic Code Generation was done, since the beginning of this century. On a second stage we present many issues related to our proposal, by detailing many tools which were used on the implementation process. Finally, on the third stage we present the achieved results by using our implementation. As it was previously mentioned, we have performed a detailed search about relevant papers for a comprehensive bibliography on Automatic Code Generation [15].

Considering the second stage, specifically during the implementation, different stages of software development were analyzed. For instance, the development of information systems, presents repetitive operations and it was recurrent from the start of software engineering, so it is convenient to use code generators that automate these tasks, so that the probability of making mistakes and therefore saving time and money can be reduced. According to studies, on average, software developers spend 50% of their programming time finding and correcting errors [16], which represents a high cost in time and effort, that can be well avoided. When software is difficult to create or modify, programmers spend most of their time concentrating on making things “work”, and less time on helping the user. But when a system is easy to work with, programmers can spend more time helping the user and less time focusing on the details of programming. Similarly, the easier it is to maintain a piece of software, the easier it is for programmers to make sure that the software is still useful [17]. When developing software applications, aside from creating efficient solutions that respond to business rules, security aspects must also be taken into account. Ensuring security is a complex process and requires knowledge of various factors, like performance and scalability. If a breach or bug is overlooked during the software testing process and goes into production, exploiting this security breach could seriously affect business continuity [13]. The proposed methodology uses proven code generation techniques, which potentially eliminates errors in the configuration process of the generated web application, this is possible because configuration parameters are also generated at the time of source code generation. The implemented application generator application uses the programming language JavaScript and JSON, which are readable, reusable and maintainable languages, much more suitable than traditional languages such as Java, Phyton and XML [18, 19].

As previously mentioned, the purpose of developing web applications that manage information and have a similar appearance to desktop applications, the ExtJs framework was used, which is based on the model, view, controller (MVC and MVVM) pattern and It was designed to make common web application tasks work like desktop application [20].

One of the advantages of the ExtJs framework is that it is based on reuse through the specialization of base classes, and its more than 140 interface components can continue to be specialized according to needs, so that we can achieve user interfaces identical to the desktop ones, not only in appearance but in functionality, which guarantees a level of confidence in the applications developed [21]. For the development of the application that generates applications, MySQL was used as the database management system, which is widely known for its simplicity and performance, but it does not mean that only MySQL can be used, actually, is possible to use any DBMS, since the communication is done by the controller, which can be implemented in any programming language, like Java, Python, PHP, Ruby, Node.Js, or any other. In this research work, for simplicity, MySQL was used, whose main objective is to be a user-friendly database.
Because the internet is the main communication tool, it has forced the implementation of web applications massively, practically reducing all types of applications that previously could only be found as desktop applications, that is, having a user interface with various functionalities [22].

For the development of the applications generator application, predefined code templates were used for the three types of scenarios: application template generation, model generation and graphical interface generation that allow operations of: create, read, update and delete records (CRUD). The figure 1 shows how the interaction of these parts is.

![Figure 1. The MVC pattern.](image)

4. Implementation

As a first step, a complete web application was developed, this is called the web applications generator application. This application is shown in the figure 2.

![Figure 2. The main interface of the applications generator application.](image)

In addition, two complete applications were implemented (one with authentication and another without). Then, the applications generator application, depending on which type of application is needed to generate, copies the entire application structure to the new application to be generated. However, internally, each one application differs in many aspects from the other. For the application with authentication module, it involves many different techniques in order to assure the data transfer and other security issues.

The second important thing was implement a module for database connecting and data model extraction. This allows interact with all the database model, tables, and relations. Furthermore, all the application depends on the database model and structure. The main interface of the
applications generator, request the user account information in order to connect to the MySQL DBMS (see the figure 2, on the left side).

Later, is need define the physical model for the database. It could be done before or do on the fly. A typical database model looks like the shown in the figure 3.

![Figure 3. A typical database physical model.](image)

The specialization or customization of each application depends on several JSON files (see figure 4), which allow to specify the specific behaviors of each application module. Some of these configurations must done by the applications generator user interface, choosing the desired options, for instance, it’s possible to choose the data manager for each fields on tables, or link a table to another as its detail (the figure 13 shows a master-detail behavior). For each field, in a table, it’s possible to choice different types, like dateselector, checkbox, comboboxes, textboxes, fileupload, multiple fileupload, email validator, password field, and other specialized types (the figure 5). All these options allows generate a complete application in a really automated way.

![Figure 4. JSON structures for some generated applications.](image)
Figure 5. Selecting the best data type manager. On the combobox case, is possible to connect a related table, in order to select the existing data.

The user interface allows to navigate for all the selected tables previously, for which is necessary define types and configure behavior options (see figure 6). Once all the tables are configured, the new application is ready to use, and depending on if was generate with authentication it could present as in the figure 7.

An authentication module is important for most applications, and the presented applications generator incorporates one. This module is integrated on the main template and is included to all generated applications when authentication is required. This module can manage users, user groups, create new users, change password internally (when the user is logged) and generate new passwords by sending a new ones to the user email when is requested.

In summary, the following sections describe the main parts of the proposed methodology.

4.1. Generation from application templates
Initially, a complete web application, with the most common features and functionalities was implemented, and it was saved as the main template. The template is composed by all the used files, including, styles, frameworks, and other resources. Then, when a new application will
be created, the applications generator clones entirely the template allowing create all the base structure of a complete application.

4.2. Models Generation
The models of the generated application are generated by using the database model, which is the main input for generate an application. This essentially contains data fields, tables and relationships, allowing the desired behavior for the application. Generally, each generated model is assigned to a single database table. The code generator connects to the database and shows the available tables to select from, as can be seen in the figure 4.

4.3. CRUDs Generation
In this code generation stage, the controller is generated with the CRUD functions and the respective configurations for build the graphic user interfaces, with interactive functionality. In the code generation interface, the option of choosing the application and the model on which you want to generate the CRUD operations is offered since an application can have several modes and behaviors.

4.4. Main application generation
The implemented applications generator, is able to generate two main application types, integrating a login and user control system or simply a system for managing information.
5. Results
In order to solve the great challenges that programmers face and to facilitate the coding process, a template-based code generator has been developed. The code generator allows you to automate repetitive coding and configuration tasks, in addition to solving abstract implementation details, reducing development time and therefore costs.

In order to show the functionalities of the proposed code generator, the development of some common information systems are presented. The application development process consists of three parts which are: main application structure generation, model generation and interface generation, which allow the different CRUD operations to be carried out. In order to enhance
the usability of the code generator, an intuitive graphical interface was developed that will guide
the developer during the generation process, as can be seen in figure 9.

5.1. Documentary management system
In a first example, a documentary management application has been implemented, since this
type of application is quite popular and there are a lot of implementations, even like as
complete software development projects. The figure 12 shows how works a manual documentary
management in real life, without a management system (software).

For this first example, firstly a database has been built to represent the business model for
the problem (see figure 10).

![E-R modeling for the documentary management database.](image)

Then, the web applications generator application is started, where it requests connect to the
database manager (see the figure 2, on the left side).

Next, as explained above, it needs select the database which the application will be built (see
the figure 8-left).

Press the “next” button and choose the option for application with user authentication (Login
module).

The documentary management application allow to see the state of each presented document,
by which offices have passed and where is now or if already was finalized and the owner have an
answer (see the figure 13). All these automatized steps save time and effort tremendously, since
any user could present a document online and can follows the response process of the document
online too.

5.2. Enrollment management system
In the second example, an application for enrollment management has been generated. This type
of system is also in great demand in various educational institutions. It includes implementation
of all aspects such as a complete software development project. The figure 14 shows the generated
application (left) and its used database (right).
5.3. Other common systems
There are many common information systems that are in demand, and they are almost always developed as independent software development projects, however, in their structure, they share many characteristics of information systems, such as CRUDs that are directly related
to the database. Thus, another example of a fairly required information system is a system for curriculum management [24]. The applications generation system is capable of generating any information system that is based on the use of a database, except that in some cases it may requires manage graphics, specific data visualization or any different type of functionality. Other common examples of information systems are: attendance control system, evaluation system, information review systems, and online tracking systems. The figure 15 shows a generated web application to register professors attendance, manage activities and the progress.

6. Discussion
Although there are some interesting tools like Genexus and WebRatio, this last is maybe the oldest and most popular, but both still are tedious to use and need to know some technical concepts including the Business Process Modeling Notation (BPMN) [25], Interaction Flow Modeling Language (IFML) [26] and Web Modeling Language (WebML) [27] and finally are the generated code is in Java or C#. These programming languages are excellent for implement a vast kind of applications, however, for medium or simple web applications are not the best tools since are complex and requires a lot of effort to achieve a good result. In that sense, our proposal is oriented for users who don’t have much technical knowledge about programming or software engineering technical concepts, where having a reasonably modeled database and with a few steps it’s possible generate a complete functional web application.
Figure 15. Web application for registration attendance, registration of activities and progress. The application shows records of a table (top). Editing/Inserting a record (bottom).

7. Conclusion and future work
With the use of the application generator it was possible to obtain web applications with complete functionalities, as if they had been developed as specific software projects, in addition to the interfaces are responsive in order to use on any kind of device, the most important thing is that the applications generator allows create the complete source code and required resources, which means that it can be improved and customized, and represents a drastically save of time and money when developing software projects. The applications generated with the applications generator were by using the MySQL database manager, however, the generator allows you to also use other database management systems such as: SQLite or PostgreSQL, since it uses generic functions which gives flexibility and dynamism. As future works, the aim is to add to the application generator the possibility of incorporating machine learning techniques so that it can autonomously build applications completely (without the intervention of a user), only by analyzing a database and some common behavior rules, which there in the most information systems. Then, learning these common characteristics the system must be able to create information systems automatically. Additionally, in order to be, the generated applications, more attractive, a variety of templates with different appearances could be incorporated. Finally, it would be interesting to make comparisons with other code generating frameworks to compare times, quality, functionality and experiences in interaction.
References

[1] M. Kaluža and B. Vukelic, “Comparison of front-end frameworks for web applications development,” Zbornik Veleučilišta u Rijeci, vol. 6, pp. 261–282, 01 2018.

[2] D.-P. Pop and A. Altar Samuel, “Designing an mvc model for rapid web application development,” Procedia Engineering, vol. 69, 12 2014.

[3] R. Tiwang, T. Oladunni, and W. Xu, “A deep learning model for source code generation,” in 2019 SoutheastCon, pp. 1–7, 2019.

[4] H. Chen, T. H. M. Le, and M. A. Babar, “Deep learning for source code modeling and generation: Models, applications, and challenges,” ACM Computing Surveys (CSUR), 2020.

[5] Z. Li, Y. Jiang, X. Zhang, and H. Xu, “The metric for automatic code generation,” Procedia Computer Science, vol. 166, pp. 279–286, 01 2020.

[6] M. Brambilla, M. Gasparini, S. Pavanetto, R. Cirami, and A. Marassi, “Code Generation based on IFML for the User Interfaces of the Square Kilometre Array (SKA),” ICALEPCS’19, vol. 19, 10 2019.

[7] G. Rossi, M. Urbiea, D. Distante, J. M. Rivero, and S. Firmenich, “25 Years of Model-Driven Web Engineering. What we achieved, What is missing...,” CLEI Electronic Journal, vol. 19, pp. 5 – 57, 12 2016.

[8] F. Behi and A. Kornecki, “Software development with automatic code generation: Observations from novice developer viewpoint...,” pp. 289–295, 01 2006.

[9] J. Bézivin, “On the unification power of models,” Software and Systems Modeling, vol. 4, no. 2, pp. 171–188, 2005.

[10] D. C. Schmidt, “Guest editor’s introduction: Model-driven engineering,” Computer, vol. 39, no. 2, pp. 25–31, 2006.

[11] S. Kent, “Model driven engineering,” in Integrated Formal Methods (M. Butler, L. Petre, and K. Sere, eds.), (Berlin, Heidelberg), pp. 286–298, Springer Berlin Heidelberg, 2002.

[12] L. K. Luhunu, “Survey of template-based code generation,” 2017.

[13] K. Shinde and Y. Sun, “Template-based code generation framework for data-driven software development,” pp. 55–60, 12 2016.

[14] R. Jain and U. Suman, “A project management framework for global software development,” ACM SIGSOFT Software Engineering Notes, vol. 43, pp. 1–10, 03 2018.

[15] V. Y. Rosales-Morales, G. Alor-Hernandez, J. L. Garcia-Alcaraz, R. Zatarain-Cabada, and M. L. Barron-Estrada, “An analysis of tools for automatic software development and automatic code generation,” Revista Facultad de Ingenieria Universidad de Antioquia, pp. 75 – 87, 12 2015.

[16] T. Britton, L. Jeng, G. Carver, and P. Cheak, “Reversible debugging software “quantify the time and cost saved using reversible debuggers”,” 2013.

[17] M. Kanat-Alexander, Code Simplicity: The Fundamentals of Software. O’Reilly Media, Inc., 2012.

[18] A.-R. Breje, R. Gyorodi, C. Györödi, D. Zmaranda, and G. Pecherle, “Comparative study of data sending methods for xml and json models,” International Journal of Advanced Computer Science and Applications, vol. 9, 01 2018.

[19] A. Ansari, “Analysis and performance issue of java and its framework and impacts on web application,” 02 2019.

[20] W. Han, L. Di, P. Zhao, and X. Li, “Using ajax for desktop-like geospatial web application development,” pp. 1 – 5, 09 2009.

[21] S. Kulkarni and S. Sachdeva, “Performance comparison for data storage - db4o and mysql databases,” pp. 166–170, 08 2014.

[22] J. Brito and J. L. Pereira, “From a desktop application to a web api a code conversion approach,” pp. 6–10, 03 2019.

[23] J. H. M. Castro, “Desarrollo de una aplicaci´ on web para la mejora de la gesti´ on de tr´ amite documentario en la direcci´ on regional de agricultura de madre de dios,” Master’s thesis, Universidad Inca Garcilaso de la Vega, 2 2018.

[24] L. Carvajal-Ortiz, B. Florian-Gaviria, and J. F. D´ıaz, “Models, methods and software prototype to support the design, evaluation, and analysis in the curriculum management of competency-based for higher education,” in 2019 XLV Latin American Computing Conference (CLEI), pp. 1–10, 2019.

[25] W. Anst, Business Process Modeling Notation. 01 2017.

[26] N. Laaz, K. Wakil, S. Mbarki, and D. Jawawi, “Comparative analysis of interaction flow modeling language tools,” Journal of Computer Science, vol. 14, 10 2018.

[27] S. Ceri, P. Fraternali, A. Bongio, M. Branbilla, S. Comai, and M. Matera, “Designing data-intensive web applications,” 06 2003.