UNIVERSITÀ DELLA CALABRIA

UNIVERSITA' DELLA CALABRIA

DIMES - Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

Scuola di Dottorato

Ingegneria dei Sistemi e Informatica

CICLO

XXV

TITOLO TESI

A Novel Cooperative Framework for Web 3.0 Investigating Recommendation and Process Mining Issues

Settore Scientifico Disciplinare *Ing-Inf/05 Sistemi Di Elaborazione Delle Informazioni*

Direttore: Ch.mo Prof. *Sergio Greco*

Firma

Supervisore: Ch.mo Prof. *Sergio Flesca*

Firma

Dottorando: Dott. *Antonio Bevacqua*

Firma

Antonio Bevacqua

A Novel Cooperative Framework for Web 3.0 Investigating Recommendation and Process Mining Issues

– Monograph –

November 28, 2013

Springer
Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

To Federica and Emanuele

Preface

The Web is Dead [44].

Classic web is dead. We are entering in the era of Web 3.0. And it is the future.

The Web is an evolving system which adapts itself to the users' needs, as witnessed by the introduction of Web2.0 and Web3.0. As a matter of fact, the Web2.0 and Web3.0 proposals aimed at supporting the end users in their Web browsing activities, by providing them with adequate tools.

The purpose of this work is proposing an architectural paradigm for developing content-based web applications based on cooperative interaction, whose foundations are based on the principles of the Web3.0 model. Specifically, the goal of our work is to analyze issues related to the introduction of Web 3.0 in organizations and to address the problem of designing and implementing an architecture that enables organizations to adapt well to the characteristics of Web 3.0.

In order to provide end users with an effective personalized web experience, web content delivery platforms must be able to satisfy the requirements of the web 3.0. In fact, we are entering the era of web3.0, but how did we get there?

The Web1.0 is a model that saw the Web as a large container of information provided by various types of organizations. The users had a window on the Web mainly through their own homepage. From the structural point of view, the available information was statically organized into taxonomies.

The Web2.0 model tried to overcome the static nature of the previous Web1.0 by enabling some rudimentary interactions with the end users. This has fostered the development and delivery of Web services as well as the formation of user communities, which has played a key role in the enrichment of the Web information (besides organization). Paradigmatic examples of Web2.0 applications include forums, blogs and social networks. Users share their experiences and provide an initial interpretation of semantic information through the tagging system: in this respect, the term *folksonomy* [10] [11] was coined in contraposition to the taxonomy of Web1.0.

Nowdays, the not yet definitive Web3.0 model tries to further evolve the Web into the personal *universe* of each user, thus introducing the concept of *portable personal Web*, that follows from the widespread adoption of new technologies and devices.

The idea is to generate adaptive systems, such as [12], which record and analyze users' activities, in order to define suitable user profiles, whose knowledge is in turn useful to anticipate their preferences, expectations and tastes. As a matter of fact, the single individual becomes the core of the Web, thus making the *folksonomy* evolve in the *me-onomy*.

The main goal of this thesis is the creation of a research tool supporting the solution of business and organizational problems. The context in which companies and structured organizations operate is increasingly complex, articulate, dynamic and unstable.

Indeed, nowadays companies are in a dynamic context [13], due to the globalization process. The result of this process is an increasing organizational complexity, in an increasingly competitive environment, caused by a shortening of the life cycle of products, which makes the need of innovation increase.

Therefore, companies must adopt an organizational structure for managing more flexible processes, capable to ensure continuous alignment of their business model to the market changes.

The competitive advantage is therefore less and less tied to the single component represented by the products offered. It is always connected with the ability to change and to adapt to the changing scenarios with times and costs more and more restrained.

For businesses, as a consequence, priority is to strategically manage the flow of information (documents and data) within and among business processes. This must have the same importance of production flows, and financial decision-making, and this also applies to the processes between different companies within the same value chain.

The goal of a research product is to study advanced techniques to be included in an innovative industrial scenario. The objective is to increase the productivity, efficiency and quality of the internal organizational processes.

For this reason, the single individual or organization becomes the core of the Web with its *me-onomy*.

The objective is to design a platform with the features necessary to solve the problems mentioned above.

To build a platform that provides effective solutions to the problems outlined above, it is necessary to address several research topics, which are listed below.

Specifically, in our thesis work, we have considered appropriate explore the topic of process mining. This because with the process mining techniques we are able to extract knowledge from event logs commonly available in todays information systems. These techniques provide new means to discover, monitor, and improve processes in a variety of application domains, especially in organizations.

Furthermore, in the context of our work Recommender Systems (RS) are strategically interesting. With the increasing volume of information [28], products, services (or, more generally, items) available on the Web, the role of Recommender Systems (RS) [30] and the importance of highly-accurate recommendation techniques have become a major concern also in the organizations. In particular, the goal of a RS is to provide users with not trivial recommendations, that are useful to directly experience potentially interesting items.

Finally, since the thesis is focused on the cooperation in organizations, it was extremely important to investigate, and integrate, systems for collaborative editing. In fact, it seemed appropriate to combine all the pieces of this line of research and to study how to adapt it in order to support collaborative real-time editing of complex documents by the users of our portal.

The goal then is to design and develop a platform for the management and analysis of collaborative processes and content, integrating an advanced recommendation system. The framework comes with a set of powerful tools.

From the analysis conducted and described up to this point, the framework could be considered an innovative platform if it characterized by the following features:

1. the ability to define and manage structured content;
2. integration with the paradigm of social networking and cooperation;
3. use of innovative models for data analysis and recommendation;
4. analysis of execution traces and integration with innovative features of process mining.
5. collaborative editing.

At this point, we can affirm that the framework must be equipped with different characteristics, namely:

- Content & Document Management,
- Social Cooperation,
- Workflow Management,
- Workflow Analytics.
- Recommendation Systems,
- Process Mining.
- Knowledge Extraction;
- Collaborative Editing.

These thesis is organized as follows. In the first chapter we present the state of the art, defining various problems and their current solutions. In the second chapter we define a novel cooperative framework for web 3.0, specifically addressing the issues of Content & Document Management, Social Cooperation, Workflow Management.

In the rest of work we give greater attention to three particular themes: Process Mining, Recommender Systems and Collaborative Editing. For Process Mining (third chapter) we deal with organization business processes. Our approach starts from the log analysis. Then, we build decision trees related to different execution scenarios. Any new process running case is assigned to the correspondent cluster. The prediction can be performed using the model related to the selected cluster. For recommendation systems (fourth chapter) we extend the popular Latent Dirichlet Allocation model by relaxing the bag-of-words assumption. We define three new models. The experimentation phase has proven its effectiveness. In the last part of the thesis we also present the literature on collaborative editing (fourth chapter). Finally, with great satisfaction, we present an application of what has been studied on a real case, *Condomani.it*. A product that became a reality thanks to this thesis.

During the development of this thesis, I had the privilege to work with brilliant and wonderful people. Among all, my sincere thanks go to *Sergio Flesca*, *Giuseppe Manco*, *Massimo Mazzeo*, *Luigi Pontieri*, *Domenico Saccà* for their support and guidance over the past few years. I am very grateful to them for pushing and supporting me during these years. Thanks to them I was able to study, appreciate and learn what you will find written in this work.

Rende,
November 2013 *Antonio Bevacqua*

Contents

1 State of Art 1
1.1 Introduction 1
1.2 Workflow in organizations at time of web 3.0 (with social and mobile) 2
1.3 Research issue and State of Art of Cooperative Frameworks in the context web 3.0 Era 4
1.3.1 Content & Document Management in the era of Social 5
1.3.2 Recommendation systems 6
1.3.3 Process Mining 6
1.3.4 Workflow 7
1.3.5 Collaborative Editing 8
1.3.6 Versioning 8
1.4 Management and content analysis 9
1.5 The Actual Scenario from the Point of View of Organizations 11
1.6 Aim of this Thesis 12

2 A Novel Cooperative Interaction Paradigm for Content-Based Web Applications 15
2.1 Introduction 15
2.2 Framework 16
2.2.1 Preliminaries 16
2.2.2 The Architecture of the *Borè* Platform 18
2.3 Types, Relations and Customization in Borè 21
2.3.1 Types and Relations 21
2.3.2 Customization of the *Borè* Platform 24
2.4 Behind the Scenes of the *Borè* Platform 25
2.5 Social Networking and Knowledge Discovery in *Borè* 27
2.5.1 Social Cooperations 27
2.5.2 Privileges 27
2.5.3 Collaborative Filtering 28
2.6 A Use Case 28

2.7 Related Work 31
2.8 Concluding remarks 32

3 A Data-Driven Prediction Framework for Analyzing and Monitoring Business Process Performances 33
3.1 Introduction 33
3.2 Preliminaries 35
3.3 Formal Framework 37
3.4 Learning Algorithm 40
3.5 System AA-TP (Adaptive-Abstraction Time Prediction) 42
3.6 Case Study 45
3.7 Concluding remarks 49

4 The last topics of the framework and use case 51
4.1 Probabilistic Sequence Modeling for Recommender Systems 51
4.1.1 Introduction 52
4.1.2 Modeling Sequence Data 54
4.1.3 Item Ranking 59
4.1.4 Experimental Evaluation 60
4.1.5 Concluding remarks 63
4.2 Collaborative Editing 63
4.2.1 Goal 64
4.2.2 Benefits for end users 65
4.2.3 Contribution of Working 65
4.2.4 The problem of simultaneous editing 66
4.2.5 Introduction to Operational Transformation 66
4.2.6 The model 67
4.2.7 Algorithms 69
4.2.8 Algorithm dOPT 69
4.2.9 Algorithm GOT 70
4.2.10 Algorithm GOTO 70
4.2.11 Algorithm Jupiter 71
4.2.12 Algorithm SOCT3 71
4.2.13 Algorithm Admissibility-Based 72
4.2.14 Google Wave 73
4.2.15 Integration in Caldera 73
4.2.16 Extension of the algorithm 74
4.2.17 Saving the document 76
4.2.18 Summary of an editing session 76
4.2.19 Concluding remarks 77
4.3 Caldera architecture scheme 78
4.4 Use case: Condomani.it 78
4.4.1 The idea 78
4.4.2 Users, product and service description 78
4.4.3 Problem solved and innovative elements 79

4.4.4 State of art 80

5 Conclusion 91

References 95

1

State of Art

The Web is an evolving system, which tries to adapt to the needs of users. The transition to Web2.0, and, currently, to Web3.0, are the expression of this trend: the goal is to focus on the leading role of the end user in Web browsing, which should be supported by adequate tools.

In this work, we propose an architectural paradigm for developing content-based web applications based on cooperative interaction, whose foundations are based on the principles of the model Web3.0. Specifically, the goal of our work is to analyze issues related to the introduction of Web 3.0 in organizations and to address the problem of designing and implementing an architecture that enables organizations to adapt well to the characteristics of Web 3.0.

The proposed architecture is extremely innovative in three respects. The first one is the possibility of defining, organizing, storing, querying and displaying the information as customizable objects and relations: a not-expert user can create the Web that he/she may prefer. A second aspect is the realization of social networks (Social Cooperations), which spontaneously arise, through user resource sharing. Finally, there is the possibility of analyzing users' browsing activities, through learning tools that enable the user to enrich his/her Web browsing experience with new knowledge.

Before analyzing our work, we make an overview of the state of art of systems and technologies. These, suitably combined, improved and used, will achieve our goal.

1.1 Introduction

Web-browsing models aim to pursue a challenging goal, i.e., to turn the Web environment into an intelligent and proactive setting, in which users play a key role. This is known as the *Semantic Web* [9] and involves the development of advanced Web interfaces, capable to enrich with semantics both the supplied information and the users' activities, with the ultimate goal of offering customized access and support to each individual user. Hitherto, efforts both in industry and academia towards the Semantic Web can be categorized as follows.

The Web1.0 is a model that saw the Web as a large container of information provided by various types of organizations. The users had a window on the Web mainly through their own homepage. From the structural point of view, the available information was statically organized into taxonomies.

The Web2.0 model tried to overcome the static nature of the previous Web1.0 by enabling some rudimentary interactions with the end users. This has fostered the development and delivery of Web services as well as the formation of user communities, which play a key role in the enrichment of the Web information (besides organization). Paradigmatic examples of Web2.0 applications include forums, blogs and social networks. Users share their experiences and provide an initial interpretation of semantic information through the tagging system: in this respect, the term *folksonomy* [10] [11] was coined in contraposition to the taxonomy of Web1.0.

Nowdays, the not yet definitive Web3.0 model tries to further evolve the Web into the personal *universe* of each user, thus introducing the concept of *portable personal Web*, that follows from the widespread adoption of new technologies and devices. The idea is to generate adaptive systems, such as [12], which record and analyze users' activities, in order to define suitable user profiles, whose knowledge is in turn useful to anticipate their preferences, expectations and tastes. As a matter of fact, the single individual becomes the core of the Web, thus making the *folksonomy* evolve in the *me-onomy*.

1.2 Workflow in organizations at time of web 3.0 (with social and mobile)

In this section we analyze now how the web 3.0 will have an impact on the classical nature of the organizations based on the use of workflows. Well remember that the web 3.0 is closely related to the spread of social networks and the massive use of mobile devices.

The collaborative work is a key point in the web 3.0. There are two aspects that are just as key points. Thanks to these, collaborative work, in our opinion, can exist.

The web 3.0 will be of considerable impact on business organization. This generates a large content of information that companies must and can optimally manage, in order to have good results.

These aspects are the technological and methodological approaches of Content Management and Workflow Management. Finding the right combination between these two types of systems is not simple. On the one hand, CMS are typically used in collaborative environments with little structure, in which the focus of the working group is fixed on the information, users enjoy complete freedom in the performance of their actions, and collaborative processes take place according to implicit rules of coordination . A working philosophy is instead assumed by distinctly different workflow management systems. In these, users are forced to perform tasks following the timing constraints and priorities imposed by the system. Finding a good compromise between the level of support / control and level of autonomy / flexibility in a system of support to collaborative work is a difficult challenge.

At present, all organizations, especially if large, they need to manage the flow of information and workflow. These flows follow the processes and decision-making. In a general way, this requirement is often satisfied by the Workflow Management Systems (WfMS); workflow are well defined here [17]. For more specific application requirements the WfMS have joined many other systems that have specialized functions: Software for Business Process Management (BPM) [18], Enterprise Resource Planning (ERP) [19–21]; Customer Relationship Management (CRM) [22–24], Enterprise Content Management (ECM) [25, 26].

At present Workflow-based systems dominate the theory and practice of Business Process Management (BPM). Despite is correct that today, in enterprise environment there is this dominance, the question is if the the workflow-based systems will be able to satisfy business needs in the future based on the assumption that the essential property of the enterprise of the future is agility. There is a work [27] based on this problem that shows why workflow-based systems may become obsolete in the future. And this can happen.

Infact, the current technological solutions are deficient in some contexts, such as the management of collaborative processes not rigidly structured or that evolve over time. This type is gradually acquiring a major relief. Is the case, for example, of document flow in the context of small organizations unstructured. Another example is that of flows that will ”stabilize” or evolving work in progress (for example, a large team where roles in the chain ”production - revision - distribution - release” are not rigidly defined and tend to become established in practice and change over time) or situations that can potentially initiate a more operational flows in which the user needs help in the choice of the optimal procedure (for example, in customer services or maintenance of systems).

In all these cases, a ”workflow” defined using only traditional tools can be too hard (it does not admit exceptions and becomes a ”blocker”) or much looser than necessary (it does not help you because it is permissive). In both cases, it fails the task and not effectively supports the user. Moreover, very often the definition phase of the workflow is cumbersome and the expression of the constraints is difficult. For this reason, a user who does not have specialized knowledge has a limited capacity for customization and tuning. The issue becomes even more relevant in relation to the spread, also for ”corporate”, of the applications of social cooperation / social networking. In fact, growing constantly active participants to a group of corporate social network and content that are produced and shared. Even the production of content has changed its nature: passing a predetermined content modeling, then static, the current trend allows users to dynamically define the contents of their interest (as in the philosophy of ”Web 3.0”). Finally, the use of content products must take place through mobile devices that are rapidly developing new features and increase their market penetration.

1.3 Research issue and State of Art of Cooperative Frameworks in the context web 3.0 Era

Aim of our work is to study the topic of the coming of Web 3.0 in organizations and to study, design and implement an architecture that enables organizations to adapt well to this change.

Fortunately, for every problem there is a study that allowed us to define a standard architecture to solve it. There are numerous studies in various fields. It was important to confront these studies to understand how to adapt the same concepts in our field. The focus was mainly on the study of architectures that were designed because of the proliferation of information to manage. Just to name a few has been designed an architecture for an e-learning system [40]. Similarly, the growth in the number of information available online has made productive development of an architecture for knowledge management systems based on artificial intelligence [41] If the number of information that can be collected by a system are many, is to take into consideration also the sources of such data, these are often the sensors. Has been studied an architecture for this purpose [42]. Finally, other architectures, such as [43] are designed to support large experiments at Cern.

So, the objective is to design a platform with the features necessary to solve the problems mentioned above. This is the ultimate goal, the intermediate is to give the contribution of research in some of the areas that will be this mentioned. The goal then is a platform for the management and analysis of collaborative processes and content. It Integrates an advanced recommendation system. The framework comes with a set of powerful tools.

From the analysis conducted and described up to this point, the framework can be considered an innovative platform if it will be appropriate to achieve as described below. The framework, to be innovative, forward-looking, and therefore be regarded as an innovative platform and useful for different reasons, should include: The framework is an innovative platform for different reasons, including:

1. the ability to define and manage structured content;
2. integration with the paradigm of social networking and cooperation;
3. use of innovative models for data analysis and recommendation;
4. analysis of execution traces and integration with innovative features of process mining.
5. collaborative editing.

Therefore we can say that: the framework should therefore provided for wide range of features:

- Content & Document Management,
- Social Cooperation,
- Workflow Management,
- Workflow Analytics.
- Recommendation Systems,
- Process Mining.

- Knowledge Extraction;
- Collaborative Editing.

In the following subsections we introduce the characteristics mentioned above. Some of these will be developed in the chapters of the thesis. In fact, it will describe the results we obtained in these areas. All innovative achievements in specific areas were then introduced and modeled within our framework.

1.3.1 Content & Document Management in the era of Social

A large number of information systems is being developed every day , these systems are designed and manufactured specifically for a single very specific context . Similarly, are designed when the context is constantly evolving. In literature [2], several methods have been developed for the development of web applications, such as [3–5].

In all these cases, the web application can manage the data that was expected at the time of design. [6] The solution to these problems was found in CMS [7, 8].

As is well defined in [1], Content Management Systems (CMS) automate the process of creating, publishing, and updating web site content. They make maintaining and updating the content of a web site easier, giving the content contributors, not just the web team, the means with which to manage their own content. They are usually made up of a front-end editor for inputting content, a back-end system for storing the content, and a template mechanism to get the content onto the web site. In the literature [2], it has been also studied how to implement a proper CMS. This also applies to the DMS, document management system.

We also analyzed the products used by enterprises. These can be divided into two broad categories, namely, frameworks and applications. Frameworks include Web infrastructures such as Django[1], Ruby on Rails [2] and Symfony [3]. These are meant to support expert users in the development of applications based on the Model-View-Controller pattern. Applications are instead content-management systems, such as Joomla! [4], Drupal [5] and Alfresco [6], whose goal is allow (even inexpert) users to simply publish their contents on the Web through a wide variety of templates, components and tools. The comparison with these tools will be analyzed in detail in the section 2.7

But the argument that we want to address is not how to create the correct CMS. Our aim is how an organization can get its custom cms without any implementation.

One of the basic ideas of us work e is that the user should be free to create its own Web universe and share it with other users. The user is the center of the web: even the user is a Web resource. Let us suppose that a user creates several Web applications

[1] https://www.djangoproject.com/
[2] http://rubyonrails.org/
[3] http://www.symfony.com/
[4] http://www.joomla.org/
[5] http://drupal.org/
[6] http://www.alfresco.com/

and publishes them to other users. An environment of social cooperation and social networking spontaneously arises if such applications share resources, since different users interact with one another. In this chapter we have discussed the rise of social in the organizations. Therefore, we addressed this issue in a comprehensive way within the Chapter 2.

1.3.2 Recommendation systems

[28] With the increasing volume of information, products, services (or, more generally, items) available on the Web, the role of Recommender Systems (RS) [30] and the importance of highly-accurate recommendation techniques have become a major concern both in e-commerce and academic research. In particular, the goal of a RS is to provide users with not trivial recommendations, that are useful to directly experience potentially interesting items. Moreover, their exploitation in e-commerce can also provide more interactions between the users and the system, that can be profitably exploited for delivering more accurate recommendations. RSs are widely employed in different contexts, from music (Last.fm[7]) to books (Amazon[8]), movies (Netflix[9]) and news (Google News[10] [31]), and they are quickly changing and reinventing the world of e-commerces [32].

The research field of recommender systems is vast. The aim of our research was the development of a probabilistic framework for modeling sequential dependencies. The intention to follow this line of research comes from the study of the models in the literature and in our observation that these had limitations. This is even more true in our scope, ie multimedia systems and documentation. From this we came up with an idea. The limitation and the next. In literature it is assumed that in a recommendation system, each token and independent from the previous one. This constraint has been changed, it has been studied the case in which they are taken into consideration contextual information. It was obtained an interesting result that is introduced into the platform Caldera. For recommendation systems we extend Latent Dirichlet Allocation popular model by relaxing the bag-of-words assumption. We defined new models. The experimentation phase has proven this. This topic is discussed in section 4.1.

1.3.3 Process Mining

As well defined here [33], process mining techniques are able to extract knowledge from event logs commonly available in todays information systems. These techniques provide new means to discover, monitor, and improve processes in a variety of application domains. There are two main drivers for the growing interest in process mining. On the one hand, more and more events are being recorded, thus, providing detailed information about the history of processes. On the other hand, there is a

[7] last.fm
[8] amazon.com
[9] netflix.com
[10] news.google.com

need to improve and support business processes in competitive and rapidly changing environments.

In most real application contexts, business processes are bound to the achievement of business goals expressed in terms of performance measures (or Key Performance Indicators), which are monitored continuously at run-time. In principle, historical log data, gathered during past enactments of a process, are a valuable source of hidden information on its behavior, which can be extracted with the help of process mining techniques [90], and eventually exploited to improve the process, and meet performance-oriented goals.

For our purpose we decided to put a lot of emphasis on these aspects.

In Chapter 3 we will see why.

We have presented a new predictive process-mining approach, which fully exploits context information, and manages to find the right level of abstraction on log traces in data-driven way. Combining several data mining and data transformation methods, the approach allows for recognizing different context-dependent process variants, while equipping each of them with a separate regression model.

Encouraging results obtained on a real application scenario show that the method is precise and robust enough, yet requiring little human intervention. Indeed, it suffices not to use extreme values for the support threshold to have low prediction errors, no matter of the other finer-grain parameters (i.e., $maxGap$ and $kTop$).

The technique has been integrated in a performance monitoring architecture, capable to provide managers and analysts with continuously updated performance statistics, as well as with the anticipated notification of possible SLA violations, which could be possibly prevented via suitable improvement policies.

So the idea is integrate all this work inside the us platform in order to offer advantage run-time services.

1.3.4 Workflow

As well defined here [17], a workflow consists of a sequence of connected steps where each step follows without delay or gap and ends just before the subsequent step may begin. It is a depiction of a sequence of operations, declared as work of a person or group, an organization of staff, or one or more simple or complex mechanisms. Workflow may be seen as any abstraction of real work. For control purposes, workflow may be a view of real work in a chosen aspect, thus serving as a virtual representation of actual work. The flow being described may refer to a document or product that is being transferred from one step to another. Workflows may be viewed as one primitive building block to be combined with other parts of an organisation's structure such as information silos, teams, projects, policies and hierarchies. In chapter 1.2, we have already written about the Workflow in Organizations at time of web 3.0 (with social and mobile). Our work will use the results described in the literature for the correct definition of an architecture that can take advantage of the mechanism of the workflow.

1.3.5 Collaborative Editing

[29] In recent years there have been several projects for collaborative writing. More and more communities are using them. For real-time collaborative editing refers to the technology that allows multiple users via an editor and a computer network, to be able to work simultaneously on a shared document. The software offers the user the possibility of this collaboration is called collaborative real-time editor. The concept of collaborative real-time editing is Englebart Douglas (1968). It took many years before getting the first implementations. The technical challenge is to find the most natural way in the eyes of the user to apply the changes made by other users. There are different techniques to achieve the desired effect. Many of these as ”locking” and ”turn taking” delay user actions. Others such as ”serialization”, ”causal ordering” and ”transformation” can not preserve the intention that the user had at the time of the change. The main problem is that, due to the latency of the interconnection network between the users, the propagation of changes is delayed. It follows that a user may have performed operations on a document without taking into account the changes made by other users.

In section 4.2 we analyzed the problem of collaborative editing. In fact, it seemed appropriate to combine all the pieces of this line of research and study how to adapt it in order to support collaborative real-time editing of complex documents by the users of the us portal.

1.3.6 Versioning

In the recent years, due to the emergence of new models of production based on collaboration, collaborative writing tools started to be increasingly used by various communities.

One of the fundamental scenarios in computer-supported collaborative work is the parallel modication of copies of a document, and the subsequent reintegration of the copies into a single document containing the modications [34], [34]. Document evolution is usually performed by creating a new document which explicitly details changes to specific paragraphs inside other document content. Obtaining (virtual) document versions corresponding to its state at a specific date is left to document users, who manually extract from library collections, and compose, the pieces of text needed to obtain the desired version. But this can be a very tedious and difficult task when changes are numerous [36].

There are differente tecquiniques to improve the collaboration inside che organizations.

In the literature there are case studies strange and varied. Even these, however, result useful in some organizations. Just to mention one example before afforntare our theme.

Interaction with multiple mouse pointers is becoming widespread for collocated collaboration on shared displays, but most technologies used to implement it do not support telepresent users. While web technologies are a common standard for applications that enable telepresent collaboration, they do not support the multipointer

interaction needed for collaboration in collocated settings. The authors of this paper have proposed an approach to provide multipointer interaction in web applications by adding multipointer support to the web browser and addressing pointer handling in a JavaScript framework [37].

One of the main objectives of a social network environment business is to help people to collaborate and provide support for cooperative work. These techniques have been made for Software merging where they are an essential aspect of the maintenance and evolution of large-scale software systems. There is a comprehensive survey and analysis of available merge approaches. In a passage from the survey we find the technique of our interest. A first distinction can be made between two-way and three-way merge tecquiniques. Two-way merging attempts to merge two versions of a software artifact without releying on the common ancestor from which both versions originated. With three-way merging, the information in the common ancestor i salso used during the merge process. This makes three-way merging more powerful than its two-way variant, in the sense that more conflicts can be detected [38].

In the web enterprise frameworks, in general, a document is not flat, but structured, therefore, this feature becomes more important, in fact, versioning should be performed on each individual field.

There are several tool online that we are using with these functionalities. In these tools you can also compare two different versions of the same document, and, if necessary, restore a previous version, this same functionality is present, such as GoogleDocs/GoogleDrive, Wikipedia. There is also a study [39] for the comprazione of the different instruments.

1.4 Management and content analysis

In recent years, the systems for the management of documentary information and content have taken hold in many working environments. They are an important tool to support the creation and sharing of information. They are a source of knowledge. They are useful for the effective operation and versatile collaborative processes. In this context, there is a growing trend to combine these systems with the characteristics of Social Networking, which have become an element of everyday experience of many individuals.

The above is confirmed by the appearance on the market of general-purpose platforms and even specific applications (eg, systems for the development of web portals or software projects, in particular open-source) These content management instruments have therefore simple and flexible services for communication and exchange of information (eg, instant messaging, blogs, message boards).

A crucial factor to meet a wide audience of businesses and consumers is certainly related to the ability to manage information in a versatile manner. This is achieved by adapting the data representation to the specific application domain and user preferences. To this end, you can take advantage of the advances made in recent years in the definition of modeling formalisms expressive, capable of representing both instances of content and documents both concepts more abstract level, using constructs

such as classes, associations and relationships of composition and of specialization. In fact, the versatility and power of such a formalism would allow to model a wide variety of documentary resources effectively and semi-structured information and associate them with ontological concepts and structures (taxonomies, partonomie, semantic relations) useful to obtain information through a paradigm "navigational".

In general, the current platforms to support collaborative work centered on a shared core document management, pay little attention to the modeling of information on the organizational structure below, both in terms of human resources, team and community, both of information resources, functional and instrumental, and the mechanisms for the modeling of organizational processes (eg, corresponding to project management or document management or content management). As mentioned above, in fact, these environments are characterized by a high level of autonomy, paid for with a low level of knowledge on the structure of organizational processes and with a low level of support and guidance provided to users in performing these tasks.

However, recently, there is a growing trend to equip some of the major commercial platforms of (Social) Content Management with workflow-oriented mechanisms for the specification and the management of processes related to the production of content.

For example, Microsoft Share Point 2010[11] supports the management of business processes and form document centric framework relying on the Windows Workflow Foundation, which provides some of the basic constructs to model the execution flows between the activities of processes. A second example is the Alfresco platform, which already allowed the re-use of a simple template workflow schema to handle the management of shared content (publication on a shared virtual space is subject to the review and approval by other users). Alfresco is shifting towards the introduction of general-purpose solutions for the management of business processes - specifically related to technology jBPM - to support the management of organizational processes more complex and application-dependent. In both cases, however, it detects a certain rigidity of the adopted solutions for the specification and the management of the processes. These have trouble adapting to the needs of modeling / flexible management of processes (and incremental modeling, change management, exception handling and / or violations of constraints), which distinguish scenarios of content production with a high level of autonomy of the players involved (for example, those of Social Production). Finally, should be noted as an integrated architecture for the management of information and content, provides the basis for a semantic analysis of their content. In fact, the use of techniques for the analysis of the contents from unstructured sources has a significant impact to the structuring of the information and to manage them more efficiently. This analysis may also have an impact on improving the efficiency of the processes that are using this information. The literature includes many innovative methods for the analysis of large amounts of unstructured data, for example, statistical models for analyzing topic / entity, methods for semantic annotation of content and structuring of data in ontologies, link

[11] http://office.microsoft.com/it-it/sharepoint/

analysis for the study of the connections between entities / users. The inherent difficulty of such task-related research makes it extremely difficult. This establishes the basis on the one hand for the advancement of scientific research in this area, and the other for the use of research results through a platform that will integrate the results of such searches for content management in support of organizations.

1.5 The Actual Scenario from the Point of View of Organizations

Our thesis work also has as its goal the creation of a research tool designed to solve business and organizational problems. We analyzed the known problems of the companies, and also the tools currently used. We have noticed that there is nothing really new, and for this reason we have addressed this research topic.

Globalization is a phenomenon for the companies very important. This has received much attention and been extensively debated both at general societal, institutional, cultural, market and business levels. Globalization of markets and reorganization are processes that involve changes in structures. The companies now are in a dynamic context [13], due to the globalization process.

Globalization has been identified by many experts as a new way firms organize their activities [14].

The context in which companies operate and structured organizations is increasingly complex, articulate and dynamic and unstable. This is due therefore to globalization. As a result, an increasing organizational complexity, in an increasingly competitive environment, caused by a shortening of the life cycle of products, resulting in a need to increase innovation. The aggravating the economic and financial crisis forces an increasing focus on cost reduction. In fact there is a new role of the modern corporate treasurer in a multinational company and its transformation in response to current challenges companies and treasurers face [15].

At the same time raising the expectations of end users, more and more advanced in their behaviors comparison, selection and purchase require a continuous improvement of product quality and service provided. Usability evaluation is essential to make sure that software products newly released are easy to use, efficient, and effective to reach goals, and satisfactory to users. For example, when a software company wants to develop and sell a new product, the company needs to evaluate usability of the new product before launching it at a market [16].

This context leads to a continuous redefinition of the competitive advantages of companies: the competitive advantage of today is no guarantee of success tomorrow.

The company must therefore adopt an organizational structure for processes that are more flexible, able to ensure continued alignment of its business model to market changes. The competitive advantage is therefore less and less tied to the single component represented by the products offered. It is always connected with the ability to change and adapt to the changing scenarios in the time and cost more and more content. The imperatives to defend and enhance the placement on the market today depend on the optimization of processes and the way people work. The only way to reduce the cost and increases the capacity for innovation. The automation of business

activities has led to an exponential growth of both paper and digital documents. These contain essential information for carrying out the same activities, especially for the interaction with the actors of the value chain. These are heterogeneous documents and information relating to customers, markets, suppliers, stakeholders, generated and used by different departments and business functions: from administration to sales, marketing, call centers, including external and Outsourcing. The management of this mass of documents and information is becoming a key element in corporate strategies. The goal is to increase efficiency in the conduct of business activities. The result is an increase in the company's quality and reducing costs. The aim is to make information accessible and usable by people working for the company, regardless of its physical location.

For businesses, therefore, priority is to strategically manage the flow of information (documents and data) within a business processes and between a process. This must have the same importance of production flows, and financial decision-making, and this also applies to the processes between companies within the value chain. The goal of a research product is to study advanced techniques to be included in an innovative industrial. The objective is to increase the productivity, efficiency and quality of the internal organizational processes.

1.6 Aim of this Thesis

The main result is, therefore, the creation of a platform with these characteristics, which provides the user with an environment in the style of social networks within which he can easily and flexibly manage all the activities of processes in the which is involved, he will interact also thanks to the suggestions that the system can provide. The user can benefit from the potential of the platform including access to it from a mobile device.

In later chapters we will analyze some of the issues stated.

The main output of the project is a newly developed prototype platform, called Caldera, which achieves a Document Management System (DMS) for the management and analysis of collaborative processes and documents. The platform will be equipped with a powerful and diverse set of tools, which offer a wide range of types of functionality.

The framework is

- oriented to the model of social networks and based on cooperation between groups and individuals;
- integrated with intelligent functions for data analysis and suggestion;
- able to define and manage structured content, even to the end user (including versioning, colaborative editing, sharing, etc.);
- scalable, flexible, and extensible.

The strengths are given by these keywords:

- Document / Content Management;
- Social Cooperation;
- Workflow Management;
- Collaborative Editing;
- Recommendation;
- Knowledge Extraction;
- Process Mining.

2

A Novel Cooperative Interaction Paradigm for Content-Based Web Applications

Web3.0 is evolving our life towards novel Web interaction paradigms aimed at transforming the Web into an enormous-in-size database where contents are directly available by means of appropriate Web query languages, and semantics of Web contents is meaningfully exploited during typical Web content processing activities (e.g., searching, browsing, indexing, and so forth) via artificial intelligence methodologies. All in one, Web3.0 is intended as a novel revolution for Web research.

Based on these motivations, in this part of thesis we propose *Borè*, a novel architecture that realizes these principles via Web3.0 paradigms.

Borè is the kernel of *Caldera*.

The proposed architecture is extremely innovative in three respects. The first one is the possibility of defining, organizing, storing, querying and displaying the information as customizable objects and relations: a not-expert user can create the Web that he/she may prefer. A second aspect is the realization of social networks (Social Cooperations), which spontaneously arise, through user resource sharing. Finally, there is the possibility of analyzing users' browsing activities, through learning tools that enable the user to enrich his/her Web browsing experience with new knowledge.

2.1 Introduction

Web-browsing models aim to pursue a challenging goal, i.e., to turn the Web environment into an intelligent and proactive setting, in which users play a key role. This is known as the *Semantic Web* and involves the development of advanced Web interfaces, capable to enrich with semantics both the supplied information and the users' activities, with the ultimate goal of offering customized access and support to each individual user. Hitherto, efforts both in industry and academia towards the Semantic Web can be categorized as defined in section 1.1.

In this chapter, we propose a novel architectural paradigm, referred to as *Borè*, for developing content-based Web applications based on Web3.0 principles. *Borè* is extremely innovative in three respects. Foremost, it allows to define, organize, store,

query and display the Web information as customizable objects and relations: an inexpert user can simply create the required Web. A second interesting feature of *Borè* is the possibility of directly supporting social networks (social cooperations), which spontaneously arise through user resource sharing. Finally, *Borè* allows the analysis of users' interaction with the published information by means of intelligent tools, that extract information on their interests, preferences and tastes from the observed interactions and exploit such information to customize and enrich their browsing experience.

The rest of chapter is structured as follows: Section 3.3 introduces some preliminaries and, then, proceeds to cover the architecture of the *Borè* platform. Section 2.3 discusses the customizable features in the proposed Web paradigm. Section 2.4 presents an overview of the formal language used as unified interaction protocol among all components of the *Borè* platform. Section 2.5 treats the establishment of social networks as well as the analysis of users' preference data for improved Web experience in *Borè*. Section 2.6 presents an exemplificative case study. Section 2.7 provides an overview of related works and highlights the main differences introduced by *Borè*. Finally, Section 2.8 concludes and mentions future work.

2.2 Framework

In this section we introduce some preliminary concepts as well as the anatomy of the devised *Borè* platform.

2.2.1 Preliminaries

The intuition behind *Borè* is to exploit some principles of the object-oriented programming [69, 70] in the context of Web-application development. The logic model or prototype of a Web application is viewed as a directed graph $\mathcal{G}$, whose nodes are the resources, while edges denotes the relations among pairs of such resources. By resource we mean all those entities that pertain to the Web application: a Web page, media contents, users, communities and so forth. A link between two resources, e.g., two Web pages, is a relation modeled as a direct edge from the former resource to the latter one.

By following the ideas of semantic Web systems, that exploit definition languages such as OWL[1], RDFS[2] and RDF[3], Web information can be generalized by structuring raw data in high-level content objects.

More precisely, each resource (resp. relation) is a pair $\{type, instance\}$: *type* is the general definition, i.e. the schema, of the resource (resp. relation) and it is composed by a set of $fields$, whereas *instance* is the type instantiation. A field is an atomic information, whose base type can be any among integer, real, string and so

[1] http://www.w3.org/TR/owl2-overview/
[2] http://www.w3.org/TR/rdf-schema/
[3] http://www.w3.org/RDF/

forth. The single inheritance property [69, 70] holds for both resources and relations. A resource (resp. relation) that inherits from another resource (resp. relation) acquires all of information within the latter in addition to its own. The separation in types and instances and the inheritance property bring the significant advantages of object-oriented software development in the design and implementation of Web applications. We here mention module reuse, simplified code development, ease of extension and maintenance and compactness of shared information.

Formally, the prototype $\mathcal{G}$ of a Web application in *Borè* is a tuple:

$$\mathcal{G} = \{R, E, RTT, ETT, RT, ET\}, \tag{2.1}$$

whose constituents are introduced next. R is the set of all resources corresponding to the nodes of $\mathcal{G}$. E is the set of all relations, i.e., the set of edges of $\mathcal{G}$. RTT and ETT are, respectively, the resource and the relation type taxonomies: resource and relation types are stored in two dedicated taxonomies enabling type inheritance. The root of RTT is the type *Object*, while the root of ETT is *Edge*. A possible implementation of both RTT and ETT will be exemplified in sec. 2.3.1. RT and ET are two forests of taxonomies. Each taxonomy in RT (resp. ET) is an inheritance hierarchy among resource (resp. relation) instances. RTT, ETT, RT and ET are useful constituents, that allow a simple management of Web browsing in terms of navigation operators as it will be described in sec. 2.4.

In our formalization, the interaction of a user with a Web application can be seen as a visit on the prototype $\mathcal{G}$: by clicking on a link within the current Web page (which corresponds to following a relation in the graph model of the Web application, that departs from the resource being currently experienced), the user moves to another Web page (i.e. visits another resource of the graph model).

The strengths following from the adoption of the foresaid logic model for Web applications deployed within *Borè* are manifold. First, a simplified resource management and querying. In particular, query processing in response to user interactions can take advantage of the solid results and foundations in the field of graph theory [67]. Second, the possibility to dynamically define new resources and types. Third, different Web applications can share the same schema (the set of all types) and differ only in their respective instances. Fourth, compactness and cooperation. As a matter of fact, different Web applications may share some resources, thus avoiding redundancies. Moreover, resource sharing allows the generation of social cooperations and social networks among users. Fifth, it is easy to store and analyze users' click streams, in order to understand their behavior and tastes.

To exemplify the foregoing concepts, we introduce in fig. 2.1 the graph (or logic model) of a toy Web application. The resources pertaining to the Web application (i.e., an institute, two workers and a student) are represented by nodes. Edges between nodes correspond to relations between resources. Actually, the graph in fig. 2.1 depicts an institute, with two workers and a student, that can publish news and organize events. The fields of individual nodes (resources) as well as their *id*s will be covered in sec. 2.3.1.

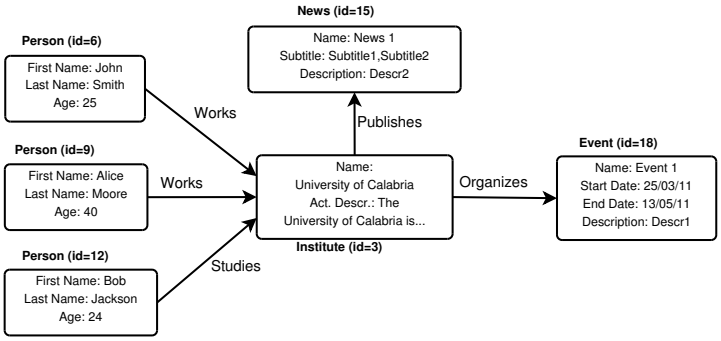


Fig. 2.1: The graph model of a toy Web application

2.2.2 The Architecture of the *Borè* Platform

The architecture of the *Borè* platform, shown in fig. 2.2, is designed around the Model-View-Controller (MVC) design pattern [68]. This allows to separate the three essential aspects of an application, i.e. its business, presentation and control logic with the purpose of considerably reducing both time and costs for development and maintenance. The architectural components of the *Borè* platform included within the Model, View and Controller layers are indicated in fig. 2.2. More specific details on the modular MVC architecture are provided below.

View

The *View* is responsible both for the navigation of the graph of a Web application and for query-result visualization. However, being in a Web environment, the final rendering of the information delivered by any Web application deployed within *Borè* is delegated to the user *browser*, which is the actual *GUI*. Instead, query answers are delegated to the *Interface Composer* module, that can communicate with the rest of the architecture.

The Interface Composer relies on some pluggable, customizable and extensible modules, that can be categorized into the following three types.

- *Template* is the module type, that specifies the whole layout of the individual Web pages.
- *Style Sheet* specifies the rendering of the different resources appearing in a Web page.
- *Custom View* is useful with all those resources, whose rendering is not performed by the two preceding modules.

The View essentially decouples the presentation of the individual resources from the layout structure of the Web pages in which they appear.

More specifically, the user interface is built automatically from the node of the graph being currently visited. Such interface displays information concerning the

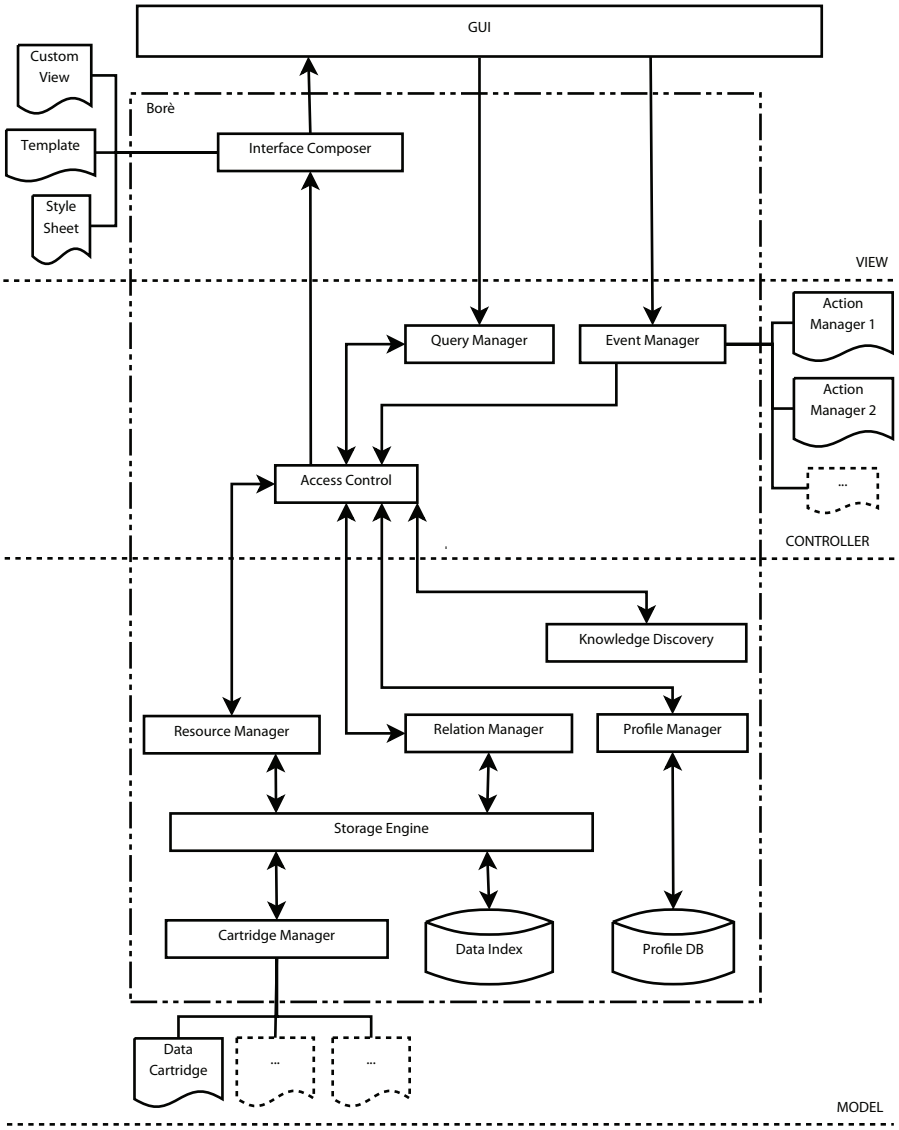


Fig. 2.2: The anatomy of the *Borè* platform

current resource and some further resources that are suitably related to the former. These latter resources are selected from those nodes of the graph of the Web application, that are connected to the current node through relations.

Controller

The *Controller* manages information processing. It is composed by the following modules:

- The *Query Manager* is the module that interprets user requests (i.e. interaction with some Web application deployed within *Borè*) and translates them into the corresponding browsing-operations or update-operations on the graph.
- The *Access Control* is essentially a message dispatcher, that coordinates nearly all modules of the *Borè* platform and controls the access of users (or groups of users) to the individual resources.
- Each resource can be associated to one or more events. The *Event Manager* module handles the events through pluggable, customizable and extensible *Action Manager*s, which differ based on the nature of the events to notify to users. Some action managers should be provided by default (e.g., timers, email notifiers and so forth).

Model

The *Model* component is the data storage layer. It offers primitive functions both for query answering and for updating the graph of the Web application as well as the taxonomies of the relative resources and relations.

Resources and relations are separately managed by the *Resource Manager* and the *Relation Manager*, respectively. The Resource Manager (resp. Relation Manager) is responsible for accessing, storing, updating and versioning the individual resources (resp. relation) and the resource (resp. relation) taxonomies.

Resource and Relation Managers communicate with the *Storage Engine*, which handles row data. Row data is retrieved through the *Data Index*, which contains meta data type information, and the *Cartridge Manager*, which loads, stores and updates the various instances by means of pluggable, customizable and extensible storage units, for instance DBs, files, etc.

The *Profile Manager* module creates resource profiles by analyzing the requests for the individual resources.

Finally, the *Knowledge Discovery* module analyzes users' interactions with the resources of a Web application for the purpose of inferring suitable profiles. These are stored and managed by the *Profile Manager* module and allow Web application to provide customized support to the individual user. Users' profiles are learnt through the application of techniques from the fields of Web Usage Mining and Artificial Intelligence to their clickstream data, in order to discover suitable behavioral patterns, that ultimately allow the customization of the browsing experience. Moreover, further techniques from the areas of Collaborative Filtering are leveraged to analyze

users' preference data and estimate their interest into not yet experienced resources. As it will be discussed in sec. 2.5, this allows the delivery of personalized suggestions to potentially interesting resources.

2.3 Types, Relations and Customization in Borè

Apart from the separation between resource and container visualization operated by the View, *Borè* also neatly separates the schema of the data from its instances through the Model of the data. Next, we provide an explanation of the data model, i.e., how data is stored and managed within *Borè*. The purpose is to provide an insight into the flexibility of data storage and management during the development of Web applications. In addition, we also deal with some further characteristics of *Borè*, that allow the customization of such an infrastructure.

2.3.1 Types and Relations

Both the resource type taxonomy RTT and the relation type taxonomy ETT introduced in sec. 3.3 are stored within the Data Index. A possible implementation of the Data Index in the context of the example of fig. 2.1 is shown in tables 2.1, 2.2 and 2.3.

Resource Types		
id	name	father_id
1	Object	null
2	Node	1
3	Community	1
4	News	2
5	Event	2
6	Person	3
7	Institute	3

Table 2.1: Resource Type table

Relation Types		
id	name	father_id
1	Edge	*null*
2	Publishes	1
3	Collaborates	1
4	Organizes	2
5	Works	3
6	Studies	3

Table 2.2: Relation Type table

Type Fields				
id	type_id	name	type	Cardinality
1	2	title	string	1.1
2	2	description	text	0.1
3	4	subtitle	string	0.n
4	5	start date	date	1.1
5	5	end date	date	0.1
6	6	first name	string	1.1
7	6	last name	string	1.1
8	6	age	int	1.1
9	7	name	string	1.1
10	7	activity description	string	0.1

Table 2.3: Resource Field table

The *Resource Type* table contains the type definitions and stores the taxonomy. A resource is identified by an internal and unique *id*, a *name* and a resource type parent, *father_id* (single inheritance): note that the root has no parent. In the proposed schema, there are seven types: Object, Node, Community, News, Event, Person and Institute, whose taxonomy is illustrated in fig. 2.3. The *Resource Field* table contains

Resource Instances		
id	type_id	father_id
1	1	*null*
2	3	1
3	7	2
4	1	*null*
5	3	4
6	6	5
7	1	*null*
8	3	7
9	6	8
10	1	*null*
11	3	10
12	6	11
13	1	*null*
14	2	13
15	4	14
16	1	*null*
17	2	16
18	5	17

Table 2.4: Resource Instance Table

Relation Instances			
id	type_id	source	dest
1	5	6	3
2	5	9	3
3	6	12	3
4	2	3	15
5	4	3	18

Table 2.5: Relation Instance Table

Int Assignments				
id	instance_id	field_id	value	pos
1	6	8	25	0
2	9	8	40	0
3	12	8	24	0

Table 2.6: Int Assignment Table

String Assignments				
id	instance_id	field_id	value	pos
1	3	9	Un. of Calabria	0
2	3	10	The Un. of Calabria is	0
3	6	6	John	0
4	6	7	Smith	0
5	9	6	Alice	0
6	9	7	Moore	0
7	12	6	Bob	0
8	12	7	Jackson	0
9	14	1	News1	0
10	15	3	Subtitle1	0
11	15	3	Subtitle2	1
12	17	1	Event1	0

Table 2.7: String Assignment Table

Text Assignments				
id	instance_id	field_id	value	pos
1	14	2	Descr1	0
2	17	2	Descr2	0

Table 2.8: Text Assignment Table

Date Assignments				
id	instance_id	field_id	value	pos
1	18	4	25/03/11	0
2	18	5	13/05/11	0

Table 2.9: Date Assignment Table

the information about the fields of a type: each field is related to a single resource and it has its own base-type and cardinality.

To better elucidate, let us consider the type with id5, whose name is *Event*. Its taxonomy path is *Object* (id1) $\rightarrow$ *Node* (id2) $\rightarrow$ *Event* (id5). The specific fields of *Event* are *start date* and *end date*. However, since *Event* inherits from *Node*, it also borrows the fields of the latter, i.e., *title* and *description*.

The definition of the *Relation Type* table is analogous to the one of the *Resource Type* table: there is an *id*, a *name* and the relation parent id (*father_id*). In the example of fig. 2.1, there are six types of relations, namely, Edge, Publishes, Collaborates, Organizes, Works and Studies, whose taxonomy is illustrated in fig. 2.4.

The separation of data from its schema in *Borè* allows to easily define a new custom type, which trivially involves to add new tuples in *Resource Type*, *Relation Type* and *Resource Field* tables.

The availability of a Data Index, that defines both the resource-type and the relation-type taxonomies (i.e., essentially the schema of Web content), allows to pop-

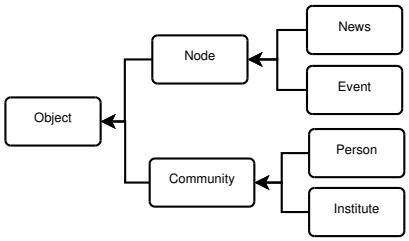


Fig. 2.3: Resource Type Taxonomy

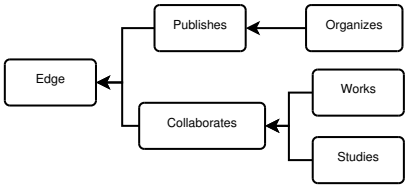


Fig. 2.4: Relation Type Taxonomy

ulate a Web application by storing the actual raw data as instances within suitable cartridges. These can be heterogenous external data sources, viewed as additional modules in the context of the architecture of the *Borè* platform. Possible examples of exploitable cartridges include relational databases, file systems, XML databases, map-reduce storage units, remote contents and so forth. A possible cartridge implementation for the context of the example in fig. 2.1 is provided by the relational database shown in tables 2.4, 2.5, 2.6, 2.7, 2.8 and 2.9.

The *Resource Instance* table stores the instances of the Web application. By looking at the attribute *type_id*, one can recognize all the resource types that form the graph of fig. 2.1, namely, an *Institute* (id 3), three *Person*s (two workers with id 6 and 9, respectively, and one student with id 12), one news (id 15) and one event (id 18).

Taxonomies are represented through the attribute *father_id*. To exemplify, the person id 12 inherits from the resource id 11 (which is a *community* instance) which, in turn, inherits from id 10, that is an *Object* instance. The fields of instances are mapped in several tables, one for each base-type. The field tables for the example of fig. 2.1 are *String Assignments*, *Int Assignments*, *Date Assignments* and *Text Assignments*. Consider the person id 12. In the string assignment table there are two entries (see the *instance_id* attribute) associated with that person, i.e., the ones with id 7 and 8, respectively. The former has *value* equals to *Bob* and its *field_id* is 6. The latter is an external key for the table *Resource Fields* (see tab. 2.3): this means that the name of the field is *first name* and its cardinality is 1.1. As a matter of fact, the first (and unique) name of person id 12 is *Bob*. The analysis of field id 8 reveals that the *last name* of the person id 12 is *Jackson*. Notice that the *pos* attribute is necessary to distinguish the different values of a field with cardinality greater than 1: a field, with more than one value, is (logically) an array, indexed by the attribute *pos*.

The *Relation Instance* table stores links between pairs of resources. Its schema includes a key attribute *id*, an external key (*type_id*) for the relation type table (see tab. 2.2) and two further external keys for the resource instance table, namely *source* and *dest*, that represent the ends of the corresponding direct edge in the graph of the Web application.

Consider the relation id 5. Its *type_id* is 4, which means that it is an *Organizes* relation. The relation source is resource id 3 (whose type is *Institute*) and the relation destination is resource id 18 (an *Event*). An illustration of this relation with related

resource taxonomies is shown in fig. 2.5. The illustration shows blocks with rounded corners that correspond to instances. These blocks are connected with irregular and dotted blocks, that provide details on the corresponding instances, i.e., their types and fields. The relationship of block containment denotes inheritance. Precisely, The *University of Calabria* (id3) is an *Institute*, which inherits from a *Community* (id2) as well as from an *Object* (id1), since *Community* (id2) in turn inherits from *Object* (id1). *University of Calabria* has two fields, namely (*Name* and *Activity Description*), and is linked to an *Event* (id18), through the relation *Organizes*. *Event*, in turn, inherits from a *Node* (id17) and from an *Object* (id16), since *Node* (id17) in turn inherits from *Object*. The fields of *Event* are *StartDate*, *EndDate*, *Name* and *Description*. Notice that both *Name* and *Description* are inherited from *Node* (id17).

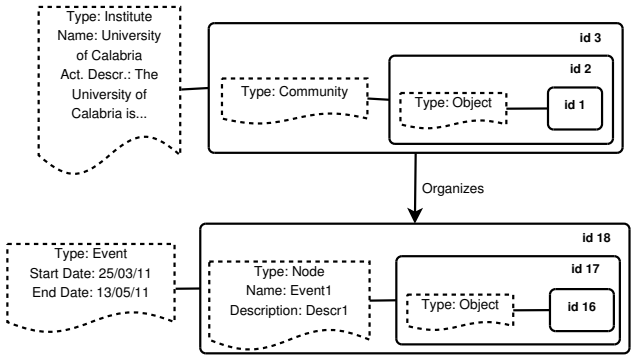


Fig. 2.5: An insight into the portion of the Web graph of fig. 2.1 including the resources *University of Calabria* and *Event 1* and their relationships

2.3.2 Customization of the *Borè* Platform

In general, enabling the dynamic creation of new types could not be sufficient in order to enrich the user browsing experience. For this reason, as it has been said earlier, the *Borè* platform allows three further degrees of customization: the Interface Composer, the Event Manager and the exploitation of cartridges for data storage. These are briefly discussed next: *(i) Interface Composer*: a user can define the preferred visualization, through pluggable custom visualizer models (as it is described in sec. 2.2.2); in particular, the user can define views of the newly created types; *(ii) Event Manager*: each resource can be associated with one or more events; the Event Manager maps resources with the specific actions to be performed while handling the individual events; actions are external plug-ins and, hence, can be suitably customized by the user; *(iii)* the last possibility of customization is the flexibility of the data-storage layer, based on custom and pluggable cartridges: the *Cartridge Manager* is the component aimed to handle cartridges

2.4 Behind the Scenes of the *Borè* Platform

The Query Manager module translates each user interaction (simple queries, i.e. click stream, and advanced queries, i.e. queries performed via a specific form) with the Web interface of the accessed Web application into a suitable mathematical set language, that is well suited to graph-based interpretation of the Web application itself. The language is the actual engine of the *Borè* platform and allows a high degree of modularity: each architectural module can be modified, reused or re-implemented without modifying other components. We next review some fundamental aspects of such language, by exploiting the definitions expressed in sec. 2.2.1.

The starting point is represented by some of its most basic operators. One such operator is *IsA*: given a type t and a taxonomy T, such that $t \in T$, *IsA* returns all the types which lie on the path from the taxonomy root to t. Formally:

$$IsA(t,T) = \{t\} \cup \{t' | \exists t' (t' \rightarrow t) \in T\} \cup \{t' | \exists t'' : (t'' \rightarrow t) \in T \wedge t' \in IsA(t'',T)\}$$

where $(x \rightarrow y)$ is a tree branch with x as the parent, while y is the child: in other words y inherits from x.

Another operator is $type$: given a resource (relation) instance i and a type taxonomy T, $type$ returns t, the last type, top-down following the hierarchy in T, i belongs to.

$$t = type(i,T)$$

The $typeList$ operator is closely related to *IsA* and $type$: given a resource (relation) instance i and RTT, $typeList$ is defined as follows:

$$typeList(i,RTT) = IsA(type(i,RTT),RTT)$$

By building on the three basic operators discussed above, it is possible to define more sophisticated operators. For instance, $filterDown$ and $filterUp$ enable user browsing in the Web application. Given a resource r, $filterDown$ permits to select all neighbor resources $\{r_1,\ldots,r_n\}$ such that a direct edge from r to r_i (where $i = 1,\ldots,n$) exists in the Web application graph.

Formally, given a resource r, two subsets $RTT' \subseteq RTT$ and $ETT' \subseteq ETT$, and a boolean function δ (that represents some generic binary property):

$$\begin{aligned} filterDown(r,RTT',ETT',\delta,RTT,ETT,R,E) = \{r' | r,r' \in R \wedge (r \Rightarrow r') \in E \\ \wedge\ IsA((r \Rightarrow r'),ETT) \cap ETT' \neq \emptyset \\ \wedge\ typeList(r',RTT) \cap RTT' \neq \emptyset \wedge r' \vdash \delta\} \end{aligned}$$

where notation $(x \Rightarrow y)$ indicates a relation from the resource x to the resource y and $r' \vdash \delta$ means that δ holds true on r'. Symmetrically, $filterUp$ allows to navigate indirect edges in the graph:

$$filterUp(r,RTT',ETT',\delta,RTT,ETT,R,E) = \{r'|r,r' \in R \wedge (r' \Rightarrow r) \in E \\ \wedge\ IsA((r' \Rightarrow r),ETT) \cap ETT' \neq \emptyset \\ \wedge\ typeList(r',RTT) \cap RTT' \neq \emptyset \wedge r' \vdash \delta\}$$

To better understand the transparent translation in *Borè* of user interactions with the Web front-end into as many corresponding queries (whose results collectively provide the new content of the Web front-end to supply to the user in response to the foresaid interactions), we next enumerate some example types of queries. These are meant for the example of fig. 2.1 and their role will be clarified in the study case of 2.6.

$$filterDown\ (\ \text{Un. of Calabria}, \{Object\}, \{Publishes\}, \\ null, RTT, ETT, R, E) = \{\text{News 1}, \text{Event 1}\} \tag{2.2}$$

This query returns all resources, whose type lists contain *Object*, that are reachable from *Un. of Calabria* through a *Publishes* relation, assuming the taxonomies RTT ETT R and E and assuming there is no constraints (i.e. δ function is null).

$$filterDown\ (\ \text{Un. of Calabria}, \{News\}, \{Edge\}, \\ null, RTT, ETT, R, E) = \{\text{News 1}\} \tag{2.3}$$

This query returns all resources, whose type lists contain *News*, that are reachable from *Un. of Calabria* through a *Edges* relation, assuming the taxonomies RTT ETT R and E and assuming there is no constraints (i.e. δ function is null).

$$filterUp\ (\ \text{Un. of Calabria}, \{Person\}, \{Collaborates\}, null, \\ RTT, ETT, R, E) = \{\text{Alice Moore}, \text{John Smith}, \text{Bob Jackson}\} \tag{2.4}$$

This query returns all resources, whose type lists contain *Person*, that reach *Un. of Calabria* through a *Collaborates* relation, assuming the taxonomies RTT ETT R and E and assuming there is no constraints (i.e. δ function is null).

$$filterUp\ (\ \text{Un. of Calabria}, \{Person\}, \{Collaborates\}, \\ age < 30, RTT, ETT, R, E) = \{\text{John Smith}, \text{Bob Jackson}\} \tag{2.5}$$

This query returns all resources, whose type lists contain *Person*, that reach *Un. of Calabria* through a *Collaborates* relation, assuming the taxonomies RTT ETT R and E and assuming that we are looking for under-30 year old people.

$$filterUp\ (\ \text{Un. of Calabria}, \{Person\}, \{Works\}, \\ age < 30, RTT, ETT, R, E) = \{\text{John Smith}\} \tag{2.6}$$

This query returns all resources, whose type lists contain *Person*, that reach *Un. of Calabria* through a *Works* relation, assuming the taxonomies RTT ETT R and E and assuming that we are looking for under-30 year old people.

2.5 Social Networking and Knowledge Discovery in *Borè*

In this section we discuss two strong points of the proposed *Borè* platform, i.e., the generation of social networks, and the analysis of user preference data for the delivery of personalized suggestions.

2.5.1 Social Cooperations

One of the basic ideas of Borè is that the user should be free to create its own Web "universe" and share it with other users. The user is the center of the web: even the user is a Web resource.

Let us suppose that a user creates several Web applications and publishes them to other users. An environment of social cooperation and social networking spontaneously arises if such applications share resources, since different users interact with one another.

Fig. 2.6(a) shows a scenario in which there are two communities. Each community has a set of resources (such a set is called *Resource Pool* in fig. 2.6(a)) and some *Shared Resources*. Both the communities are allowed to read, possibly modify the shared resources and can be notified on any updates to them.

To elucidate, in the context of the toy example of fig 2.1, the two communities can be *University of Calabria* and *Alice Moore*. Hypothetically, we can imagine that the only shared resource between the two communities is *News 1*. The latter essentially enables a social cooperation between *University of Calabria* and *Alice Moore*: a detailed explanation is in sec. 2.6.

2.5.2 Privileges

The social nature of the proposed architecture calls for a flexible management of privileges on the resources. We propose a Unix-like privilege system, where each file is assigned to an owner and a group. The difference is that in our scenario a node could be assigned to several communities, so we chose to associate to each node of the graph a rich set of privileges.

A user, for instance, can access or edit a node in the graph because he/she is the owner or because he/she is a member of the group which created the resource.

Formally, a privilege G is a tuple $G = \{\alpha, \beta, \gamma\}$, where: α is the community the resource belongs to; β is a constraint over α: G can be assigned to the sub-set of α, for which β holds; γ is the privilege type: it is the set of all actions allowed on the resource.

Each resource can have 0, 1 or more privileges, which attribute a series of available actions on the resource to a subset of users in a community. This is a very fine-grained privilege management (on a single node). In many practical cases, however, this approach might be too complex and impractical, especially in configurations with a large number of nodes. For this reason a coarse-grained configuration is appreciated. In fact, it allows to select and assign privileges for a group of nodes with the same type. For example, a company might decide to public (make visible to guests) all the

news, but to reserve other types (such as invoices) only for the members of some communities and/or their owners. This approach significantly simplifies the management of privileges, but at the same time decreases assignment flexibility. A mixed approach is, instead, in our opinion the optimal strategy. A set of coarse-grained privileges can be assigned and, where necessary it is possible to define fine-grained privileges. By turning to the previous example, the company could prohibit to any guest user the invoice querying (coarse-grained constraint), but it can give to a guest user the possibility to see his own invoices (fine-grained privileges).

2.5.3 Collaborative Filtering

The social cooperation environment leads to a really interesting aspect: a lot of information is shared among the users. This information can be used to activate knowledge discovery processes (within the *Knowledge Discovery* module) in order to infer new knowledge that is used to enrich user browsing experience.

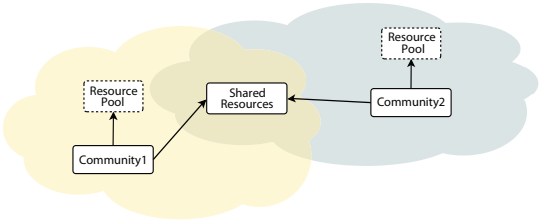
The collaborative filtering process in Borè consists of the following steps: (i) some relevant relations, among resources, are identified: recall that resources comprise individuals and communities; (ii) for each relation, a correspondence matrix is built. The number of rows in the matrix corresponds to the resources (i.e., individuals or communities), while the columns of the matrix represent all the resources involved in the chosen relation; (iii) suitable collaborative filtering techniques [73, 74] are applied on the formed matrix; (iv) finally, new relations and activities with resources, according to the discovered patterns at the previous step, are suggested to the end user.

Collaborative filtering techniques aim to find a social neighborhood (i.e. a community) of the end user, i.e., other users with similar preferences who have experienced resources not yet known to the end user. Such resources represent potentially interesting suggestions. Once the community to which the end user belongs to is identified, those resources that are expected to be mostly interesting to the her/him are recommended.

To clarify, let us consider the example relation *Collaborates* in fig. 2.6. The relative matrix is shown in fig. 2.6(b). Let us suppose that the two users r_i and r_j belong to the same preference community; in our example, both of them collaborated with r_a, r_c, r_e and r_f. Since r_i and r_j share similar tastes (they belong to the same community), and r_j also likes to collaborate r_h, probably r_i might like to collaborate with r_h too. For this reason, r_i is suggested r_h.

2.6 A Use Case

In this section we present a use case consisting of some screenshots of the Web application introduced in the toy example of fig. 2.1. The use case is aimed to highlight some major features of *Borè*. Let's suppose that the prof. *Alice Moore*, one of the workers of the institute *University of Calabria*, wishes to call a Faculty Council, provided that she has the necessary privileges.



(a) Social Cooperation

"Collaborates" Relation

	r_a	r_b	r_c	r_d	r_e	r_f	r_g	r_h	r_i
...									
r_i	yes		yes		yes	yes			
r_j	yes		yes		yes	yes		yes	

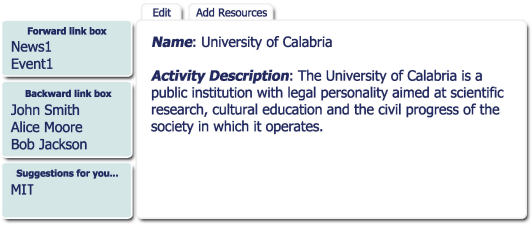
(b) Collaborative Filtering Data Matrix

Fig. 2.6: Social Cooperations and Collaborative Filtering

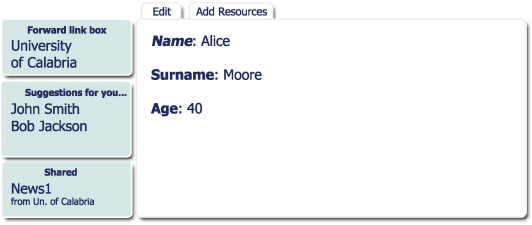
The first step is the accessing the home page of the institute (see fig. 2.7(a)). The output of *Borè* is essentially a collection of frames, each of which corresponds to a specific graphical representation of the Web contents. The main frame shows a description of a resource, whose type is *Institute* and whose realization is *University of Calabria*. The description shows the values of the fields of such a resource, i.e., *Name* and *Activity Description*. The side boxes contain the results of the $filterUp$ and $filterDown$ operators (introduced in sec. 2.4) and of the *Knowledge Discovery* module. In particular, the top frame (i.e., the *Forward Link Box*) contains the answer to the query 2.2 in sec. 2.4. The middle frame (i.e., the *Backward Link*) reports the answer to the query 2.4 in sec. 2.4. The bottom box reports the recommendation list constructed by the *Knowledge Discovery* module on behalf of the end user.

When *Alice* follows the *Alice Moore* link, her personal home page is displayed (see fig. 2.7(b)). The latter is again composed of a main frame that displays her personal information as well as a set of side boxes. The filter-up and filter-down queries to fill such side boxes are automatically issued within *Borè* when *Alice* moves from the Web page of *University of Calabria* (in fig. 2.7(a)) to her personal Web page (in fig. 2.7(b)). Notice that, in the graph model of the toy example of fig. 2.1, the resource *Alice Moore* has no incoming edges. Therefore, the filter-up operator returns an empty set and, hence, the backward link box is not visualized at all on the personal Web page. Additionally, the box named *Shared* contains all resources that *Alice Moore* shares in the Web application with *University of Calabria*. The *Shared* box actually connects *Alice Moore* to her social network.

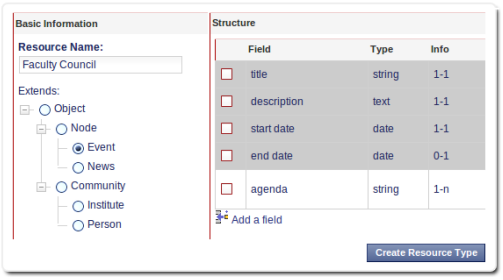
Since *Alice* wants to call a *Faculty Council*, she has to create a new resource type, and then create an instance of the new defined type. A form for the definition of a new type is shown in fig. 2.7(c). On the left side there is a tree representation of the resource type taxonomy: once selected the desired type, the new type definition will



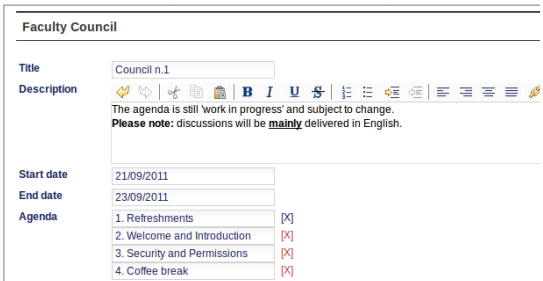
(a) Home page



(b) Personal page



(c) New type creation Page



be appended in the hierarchy. On the right side there are tools for the definition of the fields of the new type.

The form for resource instantiation is shown in fig. 2.7(d). Such a form is used by *Alice* to create a resource (i.e., an instance) of the type *Faculty Council* by providing values for each field of the type. *Alice* can also specify a connection between herself and the new resource that specifies the type of relation between them (such a relation appears in the graph model of 2.1).

Among the other operations, Alice can send an invitation to other people or, even, associate a reminder that will dispatch emails to the participants a day before the Faculty Council. In *Borè*, these operations correspond, respectively, to the addition of a new relation, say *Participates* (connecting the resources corresponding to the invited people and the council) and to the association of a custom *Action Manager* to the newly created resource *Faculty Council*.

2.7 Related Work

In this section, we discuss the main differences between *Borè* and some established competitors. Such competitors can be divided into two broad categories, namely, frameworks and applications.

Frameworks include Web infrastructures such as Django[4], Ruby on Rails [5] and Symfony [6]. These are meant to support expert users in the development of applications based on the Model-View-Controller pattern. Applications are instead content-management systems, such as Joomla! [7], Drupal [8] and Alfresco [9], whose goal is allow (even inexpert) users to simply publish their contents on the Web through a wide variety of templates, components and tools. *Borè* exhibit meaningful differences with respect to both frameworks and applications.

In particular, one general limitation of frameworks is that these are not easily customizable and extensible. As a matter of fact, the definition of new types and extensions is usually a complex process that necessarily involves expert developers. For instance, the definition of new resource types in Django involves a non trivial coding process. Moreover, the graphical tools for resource instantiation and manipulation are accessible only to system administrators. Additionally, apart from the back-office functionalities, there is no default representation of the newly defined objects for the end user. By contrast, *Borè* allows the definition of new resources and relations through user-friendly embedded graphical tools (see sec. 2.6). Also, *Borè* offers a suitable interface for type instantiation and instance manipulation, that can be simply accessed and used by both users and system administrators. Furthermore,

[4] https://www.djangoproject.com/
[5] http://rubyonrails.org/
[6] http://www.symfony-project.org/
[7] http://www.joomla.org/
[8] http://drupal.org/
[9] http://www.alfresco.com/

Borè is easily extensible through the implementation of well-defined interfaces (see sec. 2.3). Applications mainly provide the user with friendly tools for template definition. Unfortunately, apart from some exceptions such as Drupal, such competitors generally do not allow the definition of new types and, also, lack of functionalities for resource and relation manipulation in terms of taxonomies, that are instead provided by *Borè*. Therein, we recall that the latter is equipped with suitable user-friendly graphical tools for taxonomy management presented in sec. 2.3 and exemplified in sec. 2.6. Additionally, applications do not support relation customization and manipulation, that are instead a strong point of *Borè*.

To summarize, *Borè* is a platform whose features not only comprise the characteristics of both frameworks and applications, but also include new and more advanced functionalities, such as *(i)* the management of user- and community-interactions through social cooperations (resource sharing) and *(ii)* the incorporation of intelligent tools for knowledge discovery from Web and preference data, with which to provide the users with personalized lists of recommendations to potentially interesting resources.

2.8 Concluding remarks

In this part of thesis we have proposed *Borè*, a new architectural paradigm for developing content-based web applications based on Web3.0 principles. The structure allows to reach high levels of customization, and facilitates and enriches the web browsing experience of the users.

The novelty is summarized in the following list: (i)definition of new web resources; (ii) definition of mechanism of viewing, querying and storing resources; (iii) definition of events and actions associated to the resources; (iv) generation of social networks; (v) analysis of resource and relation data for processes of knowledge discovery.

Borè is extremely innovative in three respects. Foremost, it allows to define, organize, store, query and display the Web information as customizable objects and relations: a inexpert user can simply create the required Web. A second interesting feature of *Borè* is the possibility of directly supporting social networks (Social Cooperations), which spontaneously arise through user resource sharing. Finally, *Borè* allows the analysis of users' interaction with the published information by means of intelligent tools, that extract information on their interests, preferences and tastes from the observed interactions and exploit such information to customize and enrich their browsing experience. In the next chapter you can find how other feautures of the architecture.

3

A Data-Driven Prediction Framework for Analyzing and Monitoring Business Process Performances

This chapter presents a framework for analyzing and predicting the performances of a business process, based on historical data gathered during its past enactments.

This framework allow to the organization that use Caldera to analyze the process at runtime.

The framework hinges on an inductive-learning technique for discovering a special kind of predictive process models, which can support the run-time prediction of a given performance measure (e.g., the remaining processing time/steps) for an ongoing process instance, based on a modular representation of the process, where major performance-relevant variants of it are modeled with different regression models, and discriminated on the basis of context variables. The technique is an original combination of different data mining methods (ranging from pattern mining, to non-parametric regression and predictive clustering) and ad-hoc data transformation mechanisms, allowing for looking at the log traces at a proper level of abstraction, in a pretty automatic and transparent way. The technique has been integrated in a performance monitoring architecture, meant to provide managers and analysts (and possibly the process enactment environment) with continuously updated performance statistics, as well as with the anticipated notification of likely SLA violations. The approach has been validated on a real-life case study, with satisfactory results, in terms of both prediction accuracy and robustness.

3.1 Introduction

In most real application contexts, business processes are bound to the achievement of business goals expressed in terms of performance measures (or Key Performance Indicators), which are monitored continuously at run-time. In principle, historical log data, gathered during past enactments of a process, are a valuable source of hidden information on its behavior, which can be extracted with the help of process mining techniques [90], and eventually exploited to improve the process, and meet performance-oriented goals. In particular, it appears really relevant, to this regard, the recent research stream on the automated discovery of predictive process models (see,

e.g., [92,96,98]) for estimating some given performance measure over new instances of a process. The interest towards such novel mining tools stems from the observation that performance forecasts can highly improve process enactments, through, e.g., task/resource recommendations [103] or risk notification [94]. In general, these approaches try to induce some kind of prediction model for the given performance measure, based on a suitable trace abstraction function (mapping, e.g., the trace onto the set/multiset of process tasks appearing in it). For example, a non-parametric regression model is used in [96] to build the prediction for a new (possibly partial) trace, based on its similarity towards a set of historical ones – where the similarity between two traces is evaluated by comparing their respective abstract views. However, such instance-based schemes are likely to be unpractical in real environments, due to their long prediction times. A model-based prediction scheme is conversely followed in [92], where an annotated finite-state machine (AFSM) model is induced from the input log, with the states corresponding to abstract representation of log traces. The discovery of such AFSM models was combined in [98] with a context-driven (predictive) clustering approach, so that different execution scenarios can be discovered for the process, and equipped with distinct local predictors.

A key point in the induction of such models, especially in the case of complex business processes, is the definition of a suitable trace abstraction function, allowing to focus on the properties of log events that impact the more on its performance outcomes. In fact, as discussed in [92], choosing the right abstraction level is a delicate task, requiring to reasch an optimal balance between the risks of overfitting (i.e., having an overly detailed model, nearly replicating the training set, which will hardy provide accurate forecasts over unseen cases) and of underfitting (i.e., the model is too abstract and imprecise, both on the training cases and on new ones). In current approaches, the responsibility to tune the abstraction level is left to the analyst, who can select the general form of trace abstractions (e.g., set/lists of tasks), while possibly fixing a maximal horizon threshold h — i.e., only the properties of the h more recent events in a trace may appear in its abstract view. Further, FSM-like models cannot effectively exploit non-structural (context) properties of the process instances — in fact, including such data in trace abstractions would lead to a combinatorial explosion of model states.

Contribution

First of all, we try to overcome the above limitations, by devising a novel approach which can fully exploit “non structural” context data, and find a good level of abstraction over the history of process instances, in a pretty automated and transparent fashion. Our core idea is that handy (and yet accurate enough) prediction models can be learnt through various model-based regression methods (either parametric, such as, e.g., [100,102], or non-parametric, such as, e.g., [101,104]), rather than resorting to an explicit representation of process states (like in [92,98]) or to an instance-based approach (like in [96]). This clearly requires that an adequate propositional representation of the given traces be preliminary build, capturing both structural (i.e., task-related) and (“non-structural”) aspects. To this end, we convert each process trace

into a set or a multi-set of process tasks, and let the regression method decide automatically how the basic structural elements of such an abstracted trace view are to be used to make a forecast.

Leveraging the idea of [98], of combining performance prediction with a *predictive clustering* technique, we also try to equip different context-dependent execution scenarios ("variants") of the analyzed process with separate regression models. In fact, such an approach brings, in general, substantial gain in terms of readability and accuracy, besides explicitly showing the dependence of discovered clusters on context features, and speeding up (and possibly parallelize) the computation of regression models. In fact, as confirmed by the empirical results in Section 3.6, even very simple regression methods could furnish robust and accurate predictions, when combined with an effective clustering procedure. Notably, the target features used in the clustering (where context variable conversely act as descriptive attributes) are derived from frequent structural patterns (still defined as sets or bags of tasks), rather than directly using the abstract representations extracted by the log, as done in [98]. Notably, such an approach frees the analyst from the burden of explicitly setting the abstraction level (i.e., the size of patterns, in our case), which is determined instead in a data-driven way.

Finally, we devise a novel Business Process Analysis architecture, where the predictive performance models obtained with the proposed learning approach are used as a basis for the provision of advanced monitoring and analysis functionalities. In particular, in this architecture, the high-level information captured in the different kinds of process models discovered (i.e., frequent execution patterns, predictive clustering models, and cluster-wise regressors), can be exploited to better comprehend how the real behavior of the process depends on processing steps and on context factors. Moreover, such models can be reused to deploy advanced forecast services, capable to estimate, at run-time, the performance outcomes of new process instances, as well as to notify likely violations of Service Level Agreements, in advance.

Organization

The remainder of chapter is structured as follows. We first introduce some preliminaries (Section 3.2), and formally define a series of key concepts (Section 3.3). The proposed learning algorithm is then illustrated in details in Section 3.4, while Section 3.5 describes the implemented system prototype. After discussing an empirical analysis conducted on a real-life case study in Section 3.6, we finally draw a few concluding remarks and future work directions in Section 3.7.

3.2 Preliminaries

Log data

As usually done in the literature, we assume that for each process instance (a.k.a "case") a *trace* is recorded, storing the sequence of *events* happened during its unfolding. Let $\mathcal{T}$ be the universe of all (possibly partial) traces that may appear in any

log of the process under analysis. For any trace $\tau \in \mathcal{T}$, $len(\tau)$ is the number of events in τ, while $\tau[i]$ is the i-th event of τ, for $i = 1 \,..\, len(\tau)$, with $task(\tau[i])$ and $time(\tau[i])$ denoting the task and timestamp of $\tau[i]$, respectively. We also assume that the first event of each trace is always associated with task A_1, acting as unique entry point for enacting the process. This comes with no loss of generality, seeing as, should the process not have such a such a unique initial task, it could be added artificially at the beginning of each trace, and associated with the starting time of the corresponding process instance.

Let us also assume that, like in [98], for any trace τ, a tuple $context(\tau)$ of data is stored in the log to keep information about the execution context of τ, ranging from internal properties of the process instance to environmental factors pertaining the state of the process enactment system. For ease of notation, let $\mathcal{A}^{\mathcal{T}}$ denote the set of all the tasks (a.k.a., activities) that may occur in some trace of $\mathcal{T}$, and $context(\mathcal{T})$ be the space of context vectors — i.e., $\mathcal{A}^{\mathcal{T}} = \cup_{\tau \in \mathcal{T}} tasks(\tau)$, and $context(\mathcal{T}) = \{context(\tau) \mid \tau \in \mathcal{T}\}$.

Further, $\tau(i]$ is the *prefix* (sub-)trace containing the first i events of a trace τ and the same context data (i.e., $context(\tau(i] = context(\tau)$), for $i = 0 \,..\, len(\tau)$.

A *log* L is a finite subset of $\mathcal{T}$, while the *prefix set* of L, denoted by $\mathcal{P}(L)$, is the set of all the prefixes of L's traces, i.e., $\mathcal{P}(L) = \{\tau(i] \mid \tau \in L \text{ and } 1 \leq i \leq len(\tau)\}$.

Let $\hat{\mu} : \mathcal{T} \to \mathbb{R}$ be an (unknown) function assigning a performance value to any (possibly unfinished) process trace. For the sake of concreteness, we will focus hereinafter on two particular instances of such a function, where the performance measure corresponds to the *remaining time* (denoted by μ_{RT}) or the *remaining steps* (denoted by μ_{RS}), i.e., the time (resp., steps) needed to finish the corresponding process enactment. In general, we assume that performance values are known for all prefix traces in $\mathcal{P}(L)$, for any given log L. This is clearly true for the two measures mentioned above. Indeed, for each trace τ, the (actual) remaining-time of $\tau(i]$ is $\hat{\mu}_{RT}(\tau(i])$ = $time(\tau[len(\tau)]) - time(\tau[i])$, while the number of remaining steps is $\hat{\mu}_{RS}(\tau(i]) = len(\tau) - i$.

A (predictive) *Process Performance Model (PPM)* is a model that can predict the unknown performance value (i.e., the remaining time/steps in our setting) of a process enactment, based on the contents of the corresponding trace. Such a model can be viewed as a function $\mu : \mathcal{T} \to \mathbb{R}$ estimating $\hat{\mu}$ all over the trace universe — including the prefix traces of all possible process instances. Learning a PPM hence amounts to solving a particular induction problem, where the training set takes the form of a log L, and the value $\hat{\mu}(\tau)$ of the target measure is known for each (sub-)trace $\tau \in \mathcal{P}(L)$. Current approaches to this problem [92, 96, 98] relies on applying some abstraction functions to process traces, capturing only those facets of the registered events that influence the most process performances, while disregarding minor details.

Trace Abstraction

An abstracted (structural) view of a trace summarizes the tasks executed during the corresponding process enactment. Two simple ways to build such a view consist in regarding the trace as a tasks' set or multiset (a.k.a. bag), as follows.

Definition 3.1 (Structural Trace Abstraction). Let $\mathcal{T}$ be a trace universe and $A_1,\ldots,A_n$ be the tasks in $\mathcal{A}^{\mathcal{T}}$. A *structural (trace-) abstraction function* $struct^{mode}: \mathcal{T} \rightarrow \mathcal{R}_{\mathcal{T}}^{mode}$ is a function mapping each trace $\tau \in \mathcal{T}$ to an an *abstract representation* $struct^{mode}(\tau)$, taken from an *abstractions' space* $\mathcal{R}_{\mathcal{T}}^{mode}$. Two concrete instantiations of the above function, denoted by $struct^{bag}: \mathcal{T} \rightarrow \mathbb{N}^n$ (resp., $struct^{set}: \mathcal{T} \rightarrow \{0,1\}^n$), are defined next, which map each trace $\tau \in \mathcal{T}$ to a bag-based (resp., set-based) representation of its structure:

(i) $struct^{bag}(\tau) = \langle count(A_1,\tau),\ldots,count(A_n,\tau)\rangle$, where $count(A_i,\tau)$ is the number of times that task A_i occurs in τ; and *(ii)* $struct^{set}(\tau) = \langle occ(A_1,\tau),\ldots,occ(A_n,\tau)\rangle$, where $occ(A_i,\tau) = \texttt{true}$ iff $count(A_i,\tau) > 0$, for $i = 1,\ldots,n$. □

The two concrete abstraction "modes" (namely, *bag* and *set*) defined above summarize any trace τ into a vector, where each component corresponds to a single process task A_i, and stores either the number of times that A_i appears in the trace τ, or (respectively) a boolean value indicating whether A_i occur in τ or not. Notice that, in principle, we could define abstract trace representations as sets/bags over another property of the events (e.g., the executor, instead of the task executed), or even over a combination of event properties (e.g., the task plus who performed it).

Example 3.2. Let us consider a real-life case study pertaining a transshipment process, also used for experiments of Section 3.6. Basically, for each container c passing through the harbor, a distinct log trace τ_c is stored, registering all the tasks applied to c, such as: moving c by means of either a straddle-carrier (MOV), swapping c with another container (SHF), and loading c onto a ship by a shore crane (OUT). Let τ be a log trace encoding a sequence $\langle e_1,e_2,e_3\rangle$ of three events such that $task(e_1) = task(e_2) = MOV$ and $task(e_3) = OUT$. With regard to the abstract trace representations of Def. 3.1, it is easy to see that $struct^{bag}(\tau) = [2,0,1]$, and
$struct^{set}(\tau) = [1,0,1]$ — where the traces are mapped into a vector space with dimensions $A_1 \equiv MOV$, $A_2 \equiv SHF$, $A_3 \equiv OUT$.

◁

The structural abstraction functions in Def. 3.1 are a subset of those used in previous approaches to the discovery of predictive process models [92, 96, 98]. To be more precise, [96] also considers the possibility to map a trace into a vector of task durations, as well as to combine multiple structural abstractions with data attributes of the traces, while the other two approaches also allow for abstracting a trace into the list of tasks appearing in it (as an alternative to bag/set -oriented abstractions).

3.3 Formal Framework

Clustering-Based PPMs

The core idea of predictive clustering approaches [93] is that, based on a suitable clustering model, predictions for new instances can be based on the cluster where

they are estimated to belong. Two kinds of features are considered for any element z in a given instance space $Z = X \times Y$: *descriptive* features and *target* features (to be predicted), denoted by $descr(z) \in X$ and $targ(z) \in Y$, respectively. Then, a *predictive clustering model* (PCM), for a given training set $L \subseteq Z$, is a function $q : X \to Y$ of the form $q(x) = p(c(x), x)$, where $c : X \to \mathbb{N}$ is a partitioning function and $p : \mathbb{N} \times X \to Y$ is a (possibly multi-target) prediction function. Clearly, whenever there are more than one target features, q encodes a multi-regression model.

Similarly to [98], we here consider a special kind of PPM model, based on a clustering of process traces. The model, described below, is in fact a predictive clustering one, where context data play the role of descriptive attributes, while the target variables are derived by certain performance values of the traces.

Definition 3.3 (Clustering-Based Performance Prediction Model (`CB-PPM`)). Let L be a log (over $\mathcal{T}$), with context features $context(\mathcal{T})$, and $\hat{\mu} : \mathcal{T} \to \mathbb{R}$ be a performance measure, known for all $\tau \in \mathcal{P}(L)$. Then a *clustering-based performance prediction model* (`CB-PPM`) for L is a pair $M = \langle c, \langle \mu_1, \ldots, \mu_k \rangle \rangle$, encoding the unknown performance function $\hat{\mu}$ through a predictive clustering model (where k is the number of clusters found for L). Specifically, c is a partitioning function, which assigns any (possibly novel) trace to one of the clusters (based on context data), while each μ_i is the *PPM* of the i-th cluster — i.e., $c : context(\mathcal{T}) \to \{1, \ldots, k\}$, and $\mu_i : \mathcal{T} \to \mathbb{R}$, for $i \in \{1, \ldots, k\}$. The performance $\hat{\mu}(\tau)$ of any (partial) trace τ is eventually estimated as $\mu_j(\tau)$, where j=$c(context(\tau))$. □

Notice that each cluster has its own PPM model, encoding how $\hat{\mu}$ depends on the structure (and, possibly, on the context) of a trace, within that cluster. The prediction for each trace is then made with the predictor of the cluster it is assigned to (by function c).

In general, such an articulated kind of PPM can be built by inducing a predictive clustering model and multiple PPMs (as the building blocks implementing c and all μ_i, respectively). In particular, in [98], the latter task is accomplished by using the method in [92], so that each cluster is eventually provided with an AFSM model. As mentioned in Section 4.1.1, in order to develop an easier-to-use and data-adaptive approach, we will not use AFSM models (which typically require a careful explicit setting of the abstraction level), and will rather employ one of the various regression methods available for propositional data. To this purpose, an ad-hoc view of the log needs to be produced, where both the context-oriented and structure-oriented (cf. Def. 3.1) features of a trace are used as descriptive attributes, whereas the target attributes are derived by projecting the trace onto a space of structural patterns. These patterns, eventually computed with an ad-hoc data mining method, are described in detail in the following.

Structural Patterns

In our setting, structural patterns are meant to capture regularities in the structure of log traces, and they correspond to (constrained) sub-sets or sub-bag of tasks appearing frequently in the log traces (viewed as well as tasks' sets/bags). More precisely,

let $mode \in \{bag, set\}$ be a given abstraction criterion, $\mathcal{T}$ be the reference trace universe, and $\mathcal{A}^{\mathcal{T}} = \{A_1, \ldots, A_n\}$ be its associated process tasks. Then, a *(structural) pattern* w.r.t. $\mathcal{T}$ and *mode* simply is an element p of the *abstractions' space* $\mathcal{R}_{\mathcal{T}}^{mode}$ – over which function $struct^{mode}$ ranges indeed. The size of p, denoted by $size(p)$, is the number of distinct tasks in p, i.e., the number of p's components with a positive value.

Having a structural pattern the same form as a (structural) trace abstraction, we can apply usual set/bag containment operators to them all. Specifically, given two elements p and p' of $\mathcal{R}_{\mathcal{T}}^{mode}$ (be them patterns or representations of entire traces), we say that p_2 *contains* p_1 (and that p_1 is contained in p_2), denoted by $p_1 \preceq p_2$, if $p_1[j] \leq p_2[j]$, for $j = 1, \leq, n$, and for $i = 1, 2$ — where $p_i[j]$ is the i-th component of vector p_i, which ranges over $\{0,1\}^n$ or $\mathbb{N}^n$ depending on the chosen abstraction mode (cf. Def. 3.1).

Since such patterns are to be eventually used for clustering purposes, we are interested in those capturing significant behavioral schemes. In particular, an important requirement is that they occur frequently in the given log (otherwise, little, low significant clusters will be found), as specified in the following notion of support.

Definition 3.4 (Pattern Support and Footprint). Let $\tau \in \mathcal{T}$ be a trace, *mode* be a given abstraction mode, and p be pattern (w.r.t. $\mathcal{T}$ and *mode*). Then, τ *supports* p, denoted by $\tau \vdash p$ if its corresponding structural abstraction contains p (i.e., $p \preceq struct^{mode}(\tau)$). In this case, a *footprint* of p on τ is subset $F = \{f_1, \ldots, f_k\} \subseteq \{1, \ldots, len(\tau)\}$ of positions in τ, such that $struct^{mode}(\langle \tau[f_1], \ldots, \tau[f_k] \rangle) = p$. Moreover, $gap(F)$ denotes the number of events in τ which correspond to none of the positions in F but appear in between a pair of matching events, i.e., $gap(F) = \max_{f_i \in F}\{f_i\} - \min_{f_i \in F}\{f_i\} - |F| + 1$. We finally denote $gap(p, \tau) = \min\ \{\{\infty\} \cup \{\ gap(F) \mid F \text{ is a footprint of } p \text{ on } \tau\ \}\}$. □

A footprint F of a pattern p, on a trace τ supporting it, identifies a subsequence of τ which contains all the elements of τ marked by F (i.e., appearing in one of the positions of F). Clearly, the structural representation of such a subsequence coincides with p. As a special case, if τ does not support p, then $gap(p, \tau)$ is infinite.

In order to focus on significant patterns, w.r.t. a given log L, one could simply set a minimal support threshold, say $minSupp \in [0, 1)$, and discard all patterns getting a support lower than $minSupp \times |L|$ by the given log traces (viewed as tasks' bags or sets). In addition, an upper threshold $maxGap \in \mathbb{N} \cup \{\infty\}$ can be specified for the maximal gap admitted between patterns and traces, in order to focus on patterns that fit well enough the actual sequencing of tasks in the latters. Both constraints can be specified through a variant of the classical support function (actually coinciding with the latter when $maxGap = \infty$), defined as follows:

$$supp^{maxGap}(p, L) = \frac{|\{\tau \in L \mid \tau \vdash p \text{ and } gap(p, \tau) < \theta + 1\}|}{|L|} \tag{3.1}$$

Let L be a log, p be a structural pattern for a given abstraction mode *mode* (`BAG` or `SET`), *minSupp* be a minimum support threshold, and *maxGap* be a max-

imum gap threshold. Then, p is a *(minSupp,maxGap)-frequent* pattern w.r.t. L if $supp^{maxGap}(p,L) \geq minSupp$. In our approach, such patterns are considered as interesting behavioral features, useful for finding a performance-relevant partitioning of the log traces.

3.4 Learning Algorithm

Figure 3.1 illustrates the main steps of our approach to the discovery of a CB-PPM model, in the form of an algorithm, named `AA-PPM Discovery`. Essentially, the problem is approached in three main phases.

In the first phase (Steps 1-5), a set of (frequent) structural patterns are extracted from the log, which are deemed to capture the main behavioral schemes of the process, as concerns the dependence of performance on the execution of tasks. To this end, after converting the structure of each (possibly partial) trace τ into an itemset (Step 2), we compute all the structural patterns (i.e., sub-sets, of various sizes) that occur frequently in the log and effectively summarize the behaviors in the log — in particular, in the case of bag abstractions, notice that any $s = struct^{bag}(\tau) \in \mathbb{N}^n$ can be represented as $\{(A_i,k_j) \mid 0 \leq i \leq n, s[i] > 0 \text{ and } 1 \leq j \leq s[i]\}$.

More precisely, we first compute the set $\{p \in \mathcal{R}_T^m \mid supp^{maxGap}(p,S_L) \geq minSupp\}$ (cf. Def.3.4), by using function `minePatterns`, which is stored in RSP — note that this set will never be empty, since (as an extreme case) at least a singleton pattern with A_1 is frequent (no matter of $minSupp$, m and $maxGap$). These patterns are then filtered by function `filterPatterns`, which selects the $kTop$ most relevant patterns among them. Notably, we can still use all the discovered patterns, by fixing $maxGap = \infty$ (no real filter is applied in this case). Both these functions are explained in details later on.

In the second phase, the selected patterns are used to associate a series of numerical variables with all traces (Step 7), and to carry out a predictive clustering of them (Step 8). To this end, a propositional view of the log, here named *log sketch*, is produced by transforming each trace into a tuple, where context properties play as are descriptive attributes and the projection onto the space of selected patterns are the target numerical features. Specifically, any selected pattern p gives rise to a target (performance) feature, such that the value $val(\tau,p)$ taken by it on any trace τ is be computed as follows: **(i)** $val(\tau,p) = \texttt{NULL}$, if $\tau \nvdash p$, or **(ii)** $val(\tau,p) = \hat{\mu}(\tau(j^*])$, where j^* is the biggest index $j \in \{1,\ldots,len(\tau)\}$ such that $\tau(j] \vdash p$. Like in [98], the clustering is computed by inducing a Predictive Clustering Tree (PCT) [93] from the log sketch (Step 8).

Finally, each cluster is equipped with a basic (not clustering-based) PPM model, by using some suitable regression method (chosen through parameter $REGR$), provided with a dataset encoding all the prefixes that can be derived from the traces assigned to the cluster. Specifically, each such prefix τ is encoded as a tuple where $context(\tau)$ and $struct^m(\tau)$ are regarded as input values, while the associated performance measurement $\hat{\mu}(\tau)$ represents the value of the numerical target variable that is to be estimated.

Input: A log L over a trace universe $\mathcal{T}$, with associated tasks $AS = A_1, \ldots, A_n$ and target performance measure $\hat{\mu}$ (known over $\mathcal{P}(L)$), an abstraction mode $m \in \{set, bag\}$ (cf. Def. 3.1), three thresholds, $minSupp \in [0,1)$, $maxGap \in \mathbb{N} \cup \{\infty\}$, and $K_{top} \in \mathbb{N}^+ \cup \{\infty\}$, and a base regression method *REGR*.
Output: A `CB-PPM` model for L (fully encoding $\hat{\mu}$ all over $\mathcal{T}$).
Method: Perform the following steps:

1 Let $context(\tau)$ be the vector of context data associated with each $\tau \in L$;
2 Build a *structural view* S_L of $\mathcal{P}(L)$, by replacing each $\tau \in \mathcal{P}(L)$ with a transaction-like representation of $struct^m(\tau)$;
3 $RSP :=$ `minePatterns`$(S_L, m, minSupp, maxGap)$;
4 $RSP :=$ `filterPatterns`$(RSP, kTop)$;
6 Let $RSP = \{p_1, \ldots, p_s\}$;
7 Build a *log sketch* P_L for L, by using both context data and *RSP*-projected performances;
8 Learn a PCT T, using $context(\tau)$ (resp., $val(\tau, p_i)$, i=1..s) as descriptive (resp., target) features for each $\tau \in L$;
9 Let $L[1], \ldots, L[k]$ denote the discovered clusters;
10 **for each** $L[i]$ **do**
11 Induce a regression model ppm_i out of $\mathcal{P}(L[i])$, using method *REGR* — regarding, for each $\tau \in \mathcal{P}(L[i])$, $context(\tau)$ and $struct^m(\tau)$ as the input values, and the performance measurement $\hat{\mu}(\tau)$ as the target value;
10 Store ppm_i as the implementation of the prediction function $\mu_i : \mathcal{T} \rightarrow M$ (for cluster i);
11 **end**
12 **return** $\langle c, \{\mu_1, \ldots, \mu_k\} \rangle$.

Fig. 3.1: **Algorithm** `AA-PPM Discovery`.

Function `minePatterns`

This function is devoted to compute a set of (*minSupp*, *maxGap*)-frequent patterns of any size — i.e., patterns getting a support score (according to Equation 3.1) equal to or higher than *minSupp*, over a transaction-oriented view of the log. Notably, the function does not require the analyst to specify the size of each pattern (differently from the horizon threshold h of previous methods), which is actually chosen automatically, in a data-driven way. However, it allows for possibly fixing a finite *maxGap* threshold for the gaps admitted between patterns and traces, in case she/he wants to keep more details on the actual tasks' sequencing. All the specified constraints are enforced along the computation, in order to shrink the amount of patterns generated, as well as the computation time. Further details are omitted for lack of space.

Function `filterPatterns`

This function is meant to select a subset of significant and useful patterns, as to make the computation of clusters more effective and more scalable, by preventing the PCT learning algorithm to work with a sparse and high-dimensional target space. Hence, we allow the analyst to ask for only keeping the *kTop* patterns that seems to discriminate the main performance profiles at best. To this end, we employ a variant of the

scoring function ϕ proposed in [98] (giving score 0 to every feature not positively correlated with context data), which gives preference to patterns ensuring a higher values of the following measures: (i) support, (ii) correlation with the context attributes and (iii) and variability of the associated performance values (i.e., $val(p,\tau)$, with τ ranging over L). Further details can be found in [98].

3.5 System AA−TP (Adaptive-Abstraction Time Prediction)

The learning approach described so far has been integrated into a prototype system, aimed at providing advanced services for the analysis and "anticipated" (i.e., predictive) monitoring of process performances. The current implementation of the system focuses on the two time-oriented process performance measures considered in this chapter: the remaining processing time (μ_{RT}), and the number of remaining processing steps (μ_{RS}).

Looking at the right part of Figure 3.2, it can be seen that the system features a three-layer conceptual architecture. The lowest level implements basic *Data Management* modules, which are responsible for storing both historical process logs and different kinds of data (e.g., structural log views and log sketches), derived from them prior to the application of the inductive learning mechanisms described in the previous section. In fact, such data, originated by past enactments of a business process, provide a basis for the automated discovery of three different kinds of knowledge about the process: (i) frequent (structural) execution patterns, (ii) predictive clustering models (capturing the existence of diverse execution scenarios for the process, as well as their correlation with major context factors), and (iii) a series of *PPM* models (one for each discovered scenario). All these pieces of knowledge, built and handled in the *Knowledge Discovery* layer, can help better comprehend and analyze the real behavior of the process, and the dependence of its performances on both processing steps and facets of the execution context. On the other hand, thanks to their predictive nature, each clustering model and its associated local *PPM*s can be reused to configure a forecasting service for the process they were discovered from, in order to estimate (at run time and step-by-step) the performance outcome of any new instance of that process. Essentially, based on a given performance prediction model, the service implements a core method, which takes as input the partial trace of an ongoing process enactment (encoded in one of the standard formats used in ProM [91]), and returns an estimate for the associated measure (i.e., the remaining processing time/steps). Further details on the implementation of knowledge discovery functionalities are described in the final part of this section.

In principle, each deployed forecasting service could be accessed, through the *Process Enactment Gateway* interface module, by any workflow engine or any other kind of enactment system, in order to improve the process at run-time, based on ad-hoc optimization policies. The same module also offers basic services for entering new log data, and for updating a performance prediction model (based on newly added data).

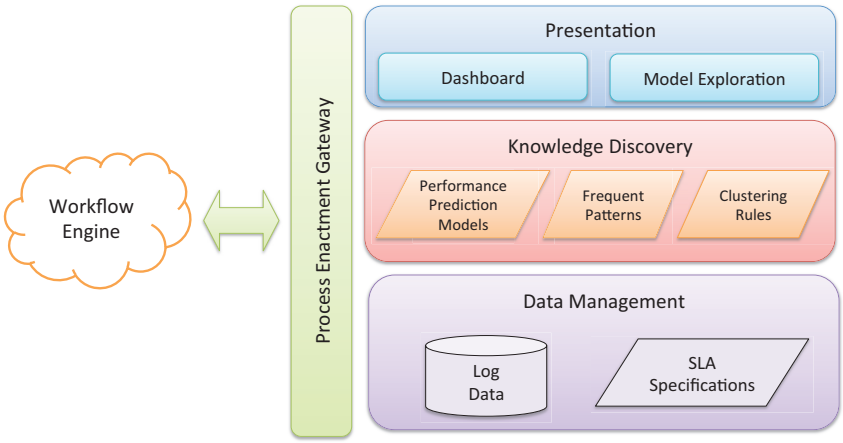


Fig. 3.2: Conceptual Architecture of the *AA-TP* System Prototype.

In addition to pure performance forecasting services, the system supports the anticipated notification of Service Level Agreement (SLA) violations, whenever a process instance is deemed as very likely to eventually fail a given quality requirement, previously established for the corresponding process (and some of its associated performance measures). The provision of such a higher-level prediction service requires that an SLA model can be defined for any process P and performance measure μ, and be kept in a suitable repository of the Data Management layer. Basically, such a model is meant to encode information about: (i) the procedure for computing the value of μ for any historical (completely executed) instance of P; (ii) a violation criterion, stating which are values that μ is not admitted to take on P's instances; (iii) the degree of sensitiveness in the the notification of possible violations. A simple approach to the specification of SLA models and associated triggering mechanisms is explained later on.

Finally, the *Presentation Layer* (still at an incomplete stage of development) is in charge of providing the user (analyst or manager) with summarized information on the past and current behavior of the analyzed business processes. In particular, it allows the user to inspect and navigate all kinds of process models and patterns extracted out of log data, as well as to create and access SLA specifications. Moreover, a customizable dashboard can be build to provide the user with aggregated statistics computed over the instances of a given process or on predefined groups of them (e.g., cases enacted for the same customer), possibly mixing predicted performance outcomes with historical ones.

SLA models

Currently the system allows for specifying a simple kind of SLA model over two types of performance measures: the total processing time and the total number of

processing steps. In particular, for each chosen process and performance measure, it is possible to set a maximum threshold M (identifying the legal range of values for the measure), and a risk tolerance threshold γ — the greater the threshold, the lower the sensitivity in detecting SLA violations (w.r.t. to the performances of the process).

Let $\tau(i]$ denote a trace, encoding the history of an ongoing process instance up to the (current) execution step i. Then, an SLA violation for $\tau(i]$ can be predicted on the basis of the likelihood $\ell_M(\tau(i])$ that the total time (or number of steps) needed to fully handle the case corresponding to τ will not exceed M (like in [94]). In more details, letting $elapsed(\tau(i])$ denote the time already spent (resp. steps already performed), this likelihood is computed as follows:

$$\ell_M(\tau) = \begin{cases} \frac{M}{elapsed(\tau(i])+\mu(\tau(i])} & \text{if } elapsed(\tau(i])+\mu(\tau(i]) > M \\ 0 & \text{otherwise} \end{cases}$$

where $\mu(\tau(i])$ is the remaining processing time (resp., nr. of steps) estimated for $\tau(i]$.

Then, an alert can be eventually triggered for the process instance associated with $\tau(i]$, if $\ell_M(\tau(i]) > \gamma$. In such a case, the user (or even the enactment system) will get aware of the fact that an SLA is likely to be infringed, so that some suitable optimization policy could be proactively put in place, in order to possibly prevent the occurrence of such a requirement violation.

Details on the Knowledge Discovery Layer

The core learning mechanism in this layer have been implemented as a plugin of the framework ProM [91]. As mentioned above, the plugin specializes algorithm `AA-PPM` *Discovery* to the case where the target performance measure coincides with the remaining processing time/steps. The plugin features the following major modules: (a) *Scenario Discovery* module, which is in charge of identifying behaviorally homogeneous groups of traces, in terms of both context data and remaining times; and (b) *Time Predictors Learning* module, which implements a range of classical regression algorithms (including, in particular, IB-k, Linear Regression, and RepTree), eventually used to induce the local predictor (i.e., PPM) of each discovered cluster — all of these predictors will compose (along with the logical rules discriminating among the clusters) the overall `CB-PPM` model, returned as main result.

More specifically, the former module consists of the following submodules: (i) the *Predictive Clustering* submodule, which groups traces sharing similar descriptive and target values, leveraging system CLUS [95] (a framework for inducing PCT models from propositional data); (ii) the *Log-View Generator* submodule, which right converts all log traces into propositional tuples, according to the ARFF format used in CLUS, relying on the explicit representation of both context data and target attributes (derived from the original log); and (iii) the *Pattern Mining* module, which exploits a transactional representation of these context-enriched traces in order to extract a set of relevant patterns out of them. In fact, these patterns are the target features used by the *Log-View Generator* in order to build a training set for the induction of a clustering model.

Table 3.1: Errors (avg±stdDev) made by AA-TP and its competitors in predicting remaining times, with different abstraction modes (namely, *bag* and *set*).

Metric	BAG				SET			
	AA-TP (IB-k)	AA-TP (RepTree)	CA-TP	AFSM	AA-TP (IB-k)	AA-TP (RepTree)	CA-TP	AFSM
rmse	0.205±0.125	**0.203±0.082**	0.291±0.121	0.505±0.059	0.287±0.123	0.286±0.084	0.750±0.120	0.752±0.037
mae	**0.064±0.058**	0.073±0.033	0.142±0.071	0.259±0.008	0.105±0.061	0.112±0.035	0.447±0.077	0.475±0.009
mape	**0.119±0.142**	0.189±0.136	0.704±0.302	0.961±0.040	0.227±0.131	0.267±0.060	2.816±0.303	2.892±0.206

Table 3.2: Errors (avg±stdDev) made by AA-TP and its competitors in predicting remaining steps, with different abstraction modes (namely, *bag* and *set*).

Metric	BAG				SET			
	AA-TP (IB-k)	AA-TP (RepTree)	CA-TP	AFSM	AA-TP (IB-k)	AA-TP (RepTree)	CA-TP	AFSM
rmse	**0.192±0.014**	0.209±0.010	0.241±0.042	0.278±0.022	0.193±0.013	0.211±0.018	0.286±0.056	0.322±0.016
mae	**0.082±0.004**	0.102±0.007	0.127±0.039	0.151±0.021	0.082±0.008	0.103±0.028	0.172±0.051	0.201±0.013
mape	**0.035±0.003**	0.042±0.003	0.060±0.025	0.056±0.016	0.035±0.003	0.043±0.030	0.094±0.036	0.110±0.008

3.6 Case Study

This section illustrates a series of experimental activities that we conducted, with the prototype system AA-TP, on some logs of a real transshipment system (already mentioned in Example 3.2), keeping trace of major logistic activities applied a sample of 5336 containers, which passed through the system in the first third of year 2006. Essentially, each container is unloaded from a ship and temporarily placed near to the dock, until it is carried to some suitable yard slot for being stocked. Symmetrically, at boarding time, the container is first placed in a yard area close to the dock, and then loaded on a cargo. Different kinds of vehicles can be used for moving a container, including, e.g., cranes, "straddle-carriers", and "multi-trailers". This basic life cycle may be extended with additional transfers, devoted to make the container approach its final embark point or to leave room for other ones. Several data attributes are available for each container as context data (of the corresponding process instance), including: the origin and final ports, its previous and next calls, various properties of the ship unloading it, physical features (such as, e.g., size, weight), and some information about its contents. Like in [98], we also considered a few more (environment-oriented) context features for each container: the hour (resp., day of the week, month) when it arrived, and the total number of containers that were in the port at that moment.

Considering the remaining processing time/steps as the target performance measure, we evaluated the effectiveness of predictions via three classic error metrics (computed via 10-fold cross validation): *root mean squared error (rmse)*, *mean absolute error (mae)*, and *mean absolute percentage error (mape)*. For ease of interpretation, the results reported next for the former two metrics have been normalized w.r.t. the average processing time/steps (computed over all the containers passed through the terminal).

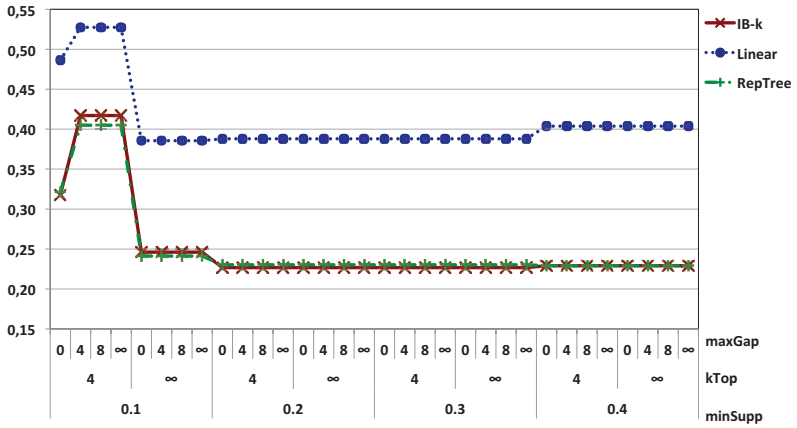


Fig. 3.3: The effect of parameters on **rmse** results.

Tuning of parameters

First of all, we tried our approach with different settings of its parameters, including the base regression method (*REGR*) for inducing the PPM of each discovered cluster. For the sake of simplicity, we here only focus on the usage of two basic regression methods: classic *Linear* regression [97], and the tree-based regression algorithm *RepTree* [104]. In addition, we consider the case where each PPM model simply encodes a *k*-NN regression procedure (denoted by *IB-k* hereinafter), as a rough term of comparison with the family of instance-based regression methods (including, in particular, the approach in [96]). For all of the above regression methods, we reused the implementations available in the popular data-mining library Weka [99].

Figure 3.3 allows for analyzing how the *rmse* scores[1] vary when using different regression methods (distinct curves are depicted for them), and different values of the parameters (namely, $maxGap \in \{0,4,8,\infty\}$, $kTop \in \{4,\infty\}$, and $minSupp \in \{0.1,\ldots,0.4\}$).

Clearly, the underlying regression method is the factor exhibiting a stronger impact on precision results. In particular, the disadvantage of using linear regression is neat (no matter of the error metrics), whereas both *IB-k* and *RepTree* methods performs quite well, and very similarly. This is good news, especially for the *RepTree* method, which is to be preferred to *IB-k* for scalability reasons. Indeed, this latter may end up being too time-consuming at run-time, when a large set of example traces must be kept – even though, differently from pure instance-based methods (like [96]), we do not need to search across the whole log, but only within a single cluster (selected via context data).

As to the remaining parameters, we notice that poorer results are obtained when $minSupp = 0.1$ and $kTop = 4$, as well as when $minSupp = 0.4$. As a matter of fact,

[1] Notice that similar trends of behavior were discovered for the *mae* and *mape* metrics.

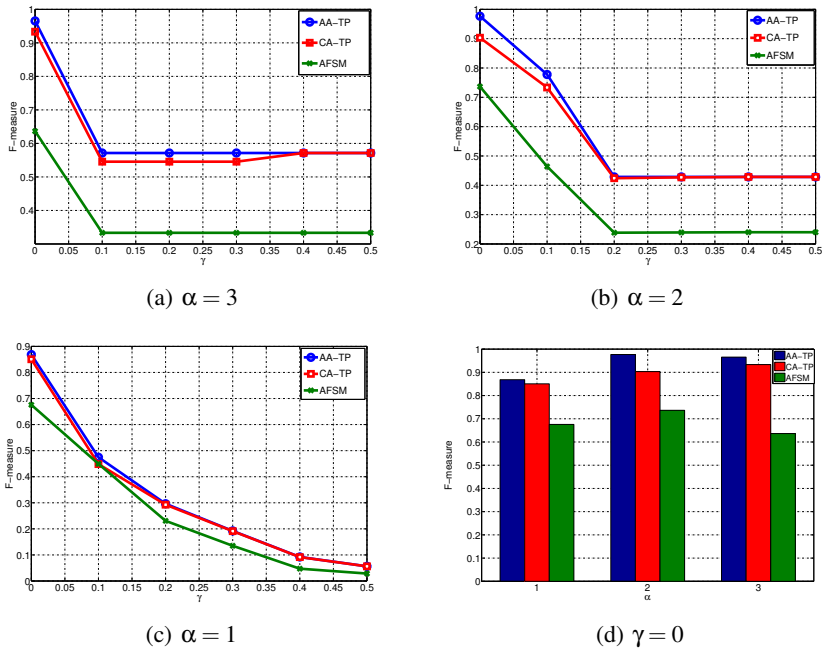
the former case epitomizes the cases where we cut little (according to frequency) during the generation of patterns, while trying to reduce their number in the filtering phase; the latter, instead, is an opposite situation where a rather high threshold support threshold is employed, at a higher risk of loosing important pieces of information on process behaviour. Remarkably, when *minSupp* gets a value from $[0.2, 0.3]$, the remaining two parameters (namely $kTop$ and *maxGap*) do not seem to affect the quality of predictions at all. In practice, it suffices to choose carefully the regression method (and a middling value of *minSupp*) to ensure good and stable prediction outcomes, no matter of the other parameters – which would be, indeed, quite harder to tune in general.

Comparison with Competitors

In order to assess the effectiveness of our approach, we compared it with two other ones, defined in the literature for the discovery of a PPM: `CA-TP` [98] and `AFSM` [92]. Tables 3.1 and 3.2 report the average errors and standard deviations made by system `AA-TP`, while varying the base regression method (namely, and *IB-k* and *RepTree*), when predicting both remaining times and remaining steps, respectively. In particular, the first half of tables regards the case when *bag* representations are used for abstracting traces, while the second to the usage of *set* abstractions. These values were computed by averaging the ones obtained with different settings of the parameters *minSupp*, *kTop*, and *maxGap*. Similarly, for both of the approaches `CA-TP` and `AFSM`, we computed the average of the results obtained using different values of the history horizon parameter h (precisely, $h = 1, 2, 4, 8, 16$), and the best-performing setting for all the remaining parameters.

Interestingly, the figures in Tables 3.1 and 3.2 indicate that our approach is more accurate than both competitors, no matter which abstraction strategy is adopted. It is worth noticing that the best results (shown in bold in the tables), for all the error metrics, are obtained when `AA-TP` is used with the *bag* abstraction mode. In particular, by looking at values in Tables 3.1 it is easy to notice that, if we combine this abstraction mode with the *IB-k* regressor, `AA-TP` manages to lower the prediction error by about 65.88% on average w.r.t. `CA-TP`, and by an astonishing 77.51% w.r.t. `AFSM`, on average (w.r.t. all the error metrics). Again, good results are obtained when using *RepTree* (still with bag abstractions), where a reduction of 59.10% (resp., 73.04%) is achieved w.r.t. to `CA-TP` (resp., `AFSM`). An even higher degree of improvement is achieved when using set-oriented abstractions. Indeed, in this case, the reduction in the average of prediction errors is of 84.58% w.r.t. `CA-TP` and of 84.97% w.r.t. `AFSM`, when using our approach with the *IB-k* regressor; moreover, when using it with *RepTree*, these reductions become 83.43% and 83.86%, respectively.

As to the prediction of remaining steps, the best outcomes are obtained when using `AA-TP` with the *IB-k* regressor (see Table 3.2). Improvements are, in general, less noticeable than those observed for the prediction of remaining times, but still substantial. Indeed, `AA-TP` manages to shrink the prediction error of 26.89% (resp. 16.49%) w.r.t. `CA-TP` and of 36.70% (resp. 27.68%) w.r.t. `AFSM`, when using the *IB-k* (resp. *RepTree*) regressor and bag abstractions. Finally, in the case of set abstractions, an error reduction of 43.77% (resp. 35.39%) w.r.t. `CA-TP`, and of 51.04%



(a) $\alpha = 3$ (b) $\alpha = 2$

(c) $\alpha = 1$ (d) $\gamma = 0$

Fig. 3.4: F-measure scores for the prediction of overtime faults by AA-TP and by baseline methods, when varying γ and the factor α.

(resp. 43.74%) w.r.t. AFSM is observed when AA-TP is used in combination with *IB-k* (resp., *RepTree*) regressors.

SLA Violation Results

We next focus on a specific kind of SLA violations, corresponding to the overcoming of maximal threshold M on the total processing time. In the context of container management systems, such a maximum processing ("dwell") time is typically set within predefined agreements, on service quality, between the shipping lines and the terminal handler. In our tests we considered a parametric specification of this threshold: $M = \alpha \times ADT$, where ADT is the average dwell time computed over all containers, and α is an integer ranging over $\{1,2,3\}$ (allowing us to simulate three different settings of the agreement level). It is important to notice that cases requiring an excessively long processing time imply high monetary costs, and detecting them in advance can allow for undertaking suitable counter-measures, possibly exploiting additional (storage/processing) resources, which are not used in normal conditions. Since, however, such a remedial policy as well come with a cost, even though it is typically lower than SLA-violation penalties, such a overtime prediction mechanism must exhibit a good trade-off between precision and recall, in order to ensure real economical the whole approach really convenient economically.

Figure 3.4 sheds light on the ability our approach discriminate "overtime" from "in-time" containers. To this purpose, we report the F-measure scores for different values of the risk threshold γ, when a fixed, good-working, configuration is used for both our approach and the baseline ones, respectively `CA-TP` [98] and `AFSM` [92]. As expected, F-measure tends to worsen when increasing γ, whatever the factor α is. Interestingly, when using lower values of γ (i.e., a more aggressive warning policy) the capability of our approach to recognize real overtime cases is ever better than the baseline predictors – in particular, a more evident improvement is obtained for $\alpha = 2$ when an astonishing F-measure of 0.98 is reached w.r.t. 0.90 and 0.74 reached by `CA-TP` and `AFSM`, respectively.

3.7 Concluding remarks

We have presented a new predictive process-mining approach, which fully exploits context information, and manages to find the right level of abstraction on log traces in data-driven way.

Combining several data mining and data transformation methods, the approach allows for recognizing different context-dependent process variants, while equipping each of them with a separate regression model.

Encouraging results obtained on a real application scenario show that the method is precise and robust enough, yet requiring little human intervention. Indeed, it suffices not to use extreme values for the support threshold to have low prediction errors, no matter of the other finer-grain parameters (i.e., $maxGap$ and $kTop$).

The technique has been integrated in a performance monitoring architecture, capable to provide managers and analysts with continuously updated performance statistics, as well as with the anticipated notification of possible SLA violations, which could be possibly prevented via suitable improvement policies.

This framework allow to the organization that use *Caldera* to analyze the process at runtime.

4

The last topics of the framework and use case

In this chapter we will discuss two important topics. The first on Reccomandation System and the second on Collaborative Editing. During the thesis work has been well addressed the first topic, among other publications are [105]. It was decided, however, not to include all of the work discussed in this thesis would otherwise have gone too far out of context from Caldera.

For users of Caldera, the benefit of the recommendation systems are enormous. In fact, when surfing on the portions of the portal (subsets of graphs made by *Borè*) they are suggested to other sections (other subsets of graphs).

Next is present the section on editing Collaborative, there we will give an overview on existing techniques and then explain how this was introduced in the Caldera.

Finally we will present a case of use of Caldera on a real platform.

4.1 Probabilistic Sequence Modeling for Recommender Systems

Probabilistic topic models are widely used in different contexts to uncover the hidden structure in large text corpora. One of the main features of these models is that generative process follows a bag-of-words assumption, i.e each token is independent from the previous one. We extend the popular Latent Dirichlet Allocation model by exploiting a conditional Markovian assumptions, where the token generation depends on the current topic and on the previous token. The resulting model is capable of accommodating temporal correlations among tokens, which better model user behavior. This is particularly significant in a collaborative filtering context, where the choice of a user can be exploited for recommendation purposes, and hence a more realistic and accurate modeling enables better recommendations. For the mentioned model we present a fast Gibbs Sampling procedure for the parameters estimation. A thorough experimental evaluation over real-word data shows the performance advantages, in terms of recall and precision, of the proposed sequence-modeling approach.

4.1.1 Introduction

Probabilistic topic models, such as the popular *Latent Dirichlet Allocation (LDA)* [78], assume that each collection of documents exhibits an hidden thematic structure. The intuition is that each document may exhibit multiple topics, where each topic is characterized by a probability distribution over words of a fixed size dictionary. This representation of the data into the latent-topic space has several advantages, as topic modeling techniques have been applied to different contexts. Example scenarios range from traditional problems (such as dimensionality reduction and classification) to novel areas (such as the generation of personalized recommendations). In most cases, LDA-based approaches have been shown to outperform state-of-art approaches.

Traditional LDA-based approaches propose a data generation process that is based on a "bag-of-words" assumption, i.e. such that the order of the items in a document can be neglected. This assumption fits textual data, where probabilistic topic models are able to detect recurrent co-occurrence patterns, which are used to define the topic space. However, there are several real-word applications where data can be "naturally" interpreted as sequences, such as biological data, web navigation logs, customer purchase history, etc. Interpreting sequence in accordance to "exchangeability", i.e., by ignoring the intrinsic sequentiality of the data within, may result in poor modeling: according to the bag-of-word assumption, co-occurrences is modeled independently for each word, via a probability distribution over the dictionary in which some words exhibit an higher likelihood to appear than others. On the other hand, sequential data may express causality and dependency, and different topics can be used to characterize different dependency likelihoods. In practice, a sequence expresses a **context** which provides valuable information for a more refined modeling.

The above observation is particularly noteworthy when data expresses preferences made by users, and the ultimate objective is to model a user's behavior in order to provide accurate recommendations. The analysis of the sequential patterns has important applications in modern recommender systems, which are always more focused on an accurate balance between personalization and contextualization techniques. For example, in Internet based streaming services for music or video (such as Last.fm[1] and Videolectures.net[2]), the context of the user interaction with the system can be easily interpreted by analyzing the content previously requested. The assumption here is that the current item (and/or its genre) influences the next choice of the user.

Recommender systems have greatly benefited from probabilistic modeling techniques based on LDA. Recent works in fact have empirically shown that probabilistic latent topics models represent the state-of-the art in the generation of accurate personalized recommendations [**?**, 75, 76]. Probabilistic techniques offer some advantages over traditional deterministic models: notably, they do not minimize a particular error metric but are designed to maximize the likelihood of the model given the data which is a more general approach; moreover, they can be used to model a

[1] `http://last.fm`

[2] `http://videolectures.net`

distribution over rating values which can be used to determine the confidence of the model in providing a recommendation; finally, they allow the possibility to include prior knowledge into the generative process, thus allowing a more effective modeling of the underlying data distribution. Notably, when preferences are implicitly modeled through selection (that is, when no rating information is available), the simple LDA best models the probability that an item is actually selected by a user [75].

A simple approach to model sequential data within a probabilistic framework has been proposed in [79]. In this work, authors present a framework based on mixtures of Markov models for clustering and modeling of web site navigation logs, which is applied for clustering and visualizing user behavior on a web site. Albeit simple, the proposed model suffers of the limitation that a single latent topic underlies all the observation in a single sequence. This approach has been overtaken by other methods based on latent semantic indexing and LDA. In [88, 89], for example, the authors propose extension of the LDA model which assume a first-order Markov chain for the word generation process. In the resulting *Bigram Model (BM)* and *Topical n-grams*, the current word depends on the current topic and the previous word observed in the sequence. The LDA Collocation Model [83] introduces a new set of random variables (for bigram status) x which denotes whether a bigram can be formed with the previous word token. The bigram status adds a more realistic than Wallach model which always generates bigrams.

Hidden Markov models [77, Chapter 13] are a general reference framework for modeling sequence data. HMMs assume that sequential data are generated using a Markov chain of latent variables, with each observation conditioned on the state of the corresponding latent variable. The resulting likelihood can be interpreted as an extension of a mixture model in which the choice of mixture components for each observation is not selected independently but depends on the choice of components for the previous observation. [84] delve in this direction, and propose an *Hidden Topic Markov Model (HTMM)* for text documents. HTTM define a Markov chain over latent topics of the document. The corresponding generative process assume that all words in the same sentence share the same topic, while successive sentences can either rely on the previous topic, or introduce a new one. The topics in a document form a Markov chain with a transition probability that depends on a binary topic transition variable ψ. When $\psi = 1$, a new topic is drawn for the n-th sentence, otherwise the same previous topic is used.

Following the research direction outlined above, in this part of thesis we study the effects of "contextual" information in probabilistic modeling of preference data. We focus on the case where the context can be inferred from the analysis of the sequence data, and we propose a topic model which explicitly makes use of dependency information for providing recommendations. As a matter of fact, the issue has been dealt with in similar works (like, e.g. [88]). Here, we resume and extend the approaches in the literature. by concentrating on the effects of such modeling on recommendation accuracy, as it explicitly reflects accurate modeling of user behavior.

In short, the contributions of this part of work can be summarized as follows.

Table 4.1: Summary of the notation used

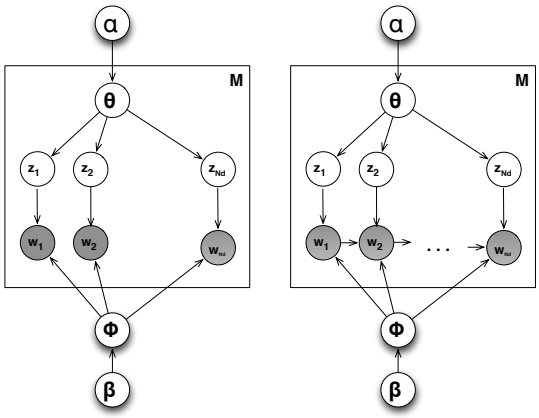
notation	*description*
M	# Users
N	# Items
K	# Topics
$\mathbf{W}$	Collection of users' traces, $\mathbf{W} = \{\overrightarrow{w}_1, \ldots, \overrightarrow{w}_M\}$
n_u	# Items in the user u's trace
$\overrightarrow{w}_u$	Item trace of user u, $\overrightarrow{w}_u = \{w_{u,1}.w_{u,2}.\cdots.w_{u,N_u-1}.w_{u,n_u}\}$
$w_{u,n}$	n-th item in the trace of user u
$\mathbf{Z}$	Collection of topic traces for each user, $\mathbf{Z} = \{\overrightarrow{z}_1, \ldots, \overrightarrow{z}_M\}$
$\overrightarrow{z}_u$	Topic trace for user u, $\overrightarrow{z}_u = \{z_{u,1}.z_{u,2}.\cdots.z_{u,N_u-1}.z_{u,n_u}\}$
$z_{u,n}$	n-th topic in the trace of user u
$n^k_{d,i}$	number of times item i has been associated with topic k for user d
$n^k_{(\cdot),i.j}$	number of times item sequence $i.j$ has been associated with topic k in $\mathbf{W}$
$n^k_{d,(\cdot)}$	number of times an item has been associated with topic k for user d
$\overrightarrow{\Theta}$	matrix of parameters $\overrightarrow{\theta}_u$
$\overrightarrow{\theta}_u$	mixing proportion of topics for the user u
$\vartheta_{u,k}$	mixing coefficient of the topic k for the user u
$\overrightarrow{\Phi}$	matrix of parameters $\overrightarrow{\varphi}_k = \{\varphi_{k,j,i}\}$
$\varphi_{k,j,i}$	mixing coefficient of the topic k for the item sequence j,i

1. We propose an unified probabilistic framework to model dependency in preference data, and instantiate the framework in accordance to a specific assumption on the sequentiality of the underlying generative process;
2. For the proposed instance, we provide the relative ranking function that can be used to generate personalized and context-aware recommendation lists;
3. We finally show that the proposed sequential modeling of preference data better models the underlying data, as it allows more accurate recommendations in terms of precision and recall.

This part of work is structured as follows. In Sec. 4.1.2 we introduce sequential modeling, and specify in Sec. 4.1.3 the corresponding item ranking functions for supporting recommendations. The experimental evaluation of the proposed approaches in then presented in Sec. 4.1.4, in which we measure the performance of the approaches in a recommendation scenario. Section 4.1.5 concludes the section with a summary of the findings and mention to further extensions.

4.1.2 Modeling Sequence Data

Let $\mathcal{U} = \{u_1, \ldots, u_M\}$ be a set of M users and $I = \{i_1, \ldots, i_N\}$ a set of N items. In the general settings, we consider a set $\mathbf{W} = \{\overrightarrow{w}_1, \ldots, \overrightarrow{w}_M\}$ of user traces, where $\overrightarrow{w}_u = \{w_{u,1}, w_{u,2}, \cdots, w_{u,n_i-1}.w_{u,n_i}\}$ is the trace of all items selected by user u in sequence. We also assume that each user action is characterized by a latent factor triggering that action. That is, a latent set $\mathbf{Z} = \{\overrightarrow{z}_1, \ldots, \overrightarrow{z}_M\}$ is associated to the data, where, again $\overrightarrow{z}_u = \{z_{u,1}, z_{u,2}, \cdots, z_{u,n_i-1}.z_{u,n_i}\}$ is a latent topic sequence, and $z_{d,n} \in \{1, \ldots, K\}$ is the latent topic associated with the item $w_{d,n} \in I$. By assuming that



(a) Latent Dirichlet Allocation (b) Token-Bigram Model

Fig. 4.1: Graphical Models

c
$\overrightarrow{d}$
$\overrightarrow{\Phi}$
$\overrightarrow{A}$ $\overrightarrow{\Phi}$
Φ
$\overrightarrow{\Phi}$
and $\overrightarrow{\Theta}$
are the distribution functions for **W** and **Z** (with respective priors $\overrightarrow{\beta}$ and $\overrightarrow{\alpha}$, we can express the complete likelihood as:

$$\begin{aligned} P(\mathbf{W},\mathbf{Z},\overrightarrow{\Theta},\overrightarrow{\Phi}|\overrightarrow{\alpha},\overrightarrow{\beta}) =& P(\mathbf{W}|\mathbf{Z},\overrightarrow{\Phi})P(\overrightarrow{\Phi}|\overrightarrow{\beta}) \\ & \cdot P(\mathbf{Z}|\overrightarrow{\Theta})P(\overrightarrow{\Theta}|\overrightarrow{\alpha}) \end{aligned} \tag{4.1}$$

where

$$P(\mathbf{W}|\mathbf{Z},\overrightarrow{\Phi}) = \prod_{d=1}^{M} P(\overrightarrow{w}_d|\overrightarrow{z}_d,\overrightarrow{\Phi})$$

$$P(\mathbf{Z}|\overrightarrow{\Theta}) = \prod_{d=1}^{M} P(\overrightarrow{z}_d|\overrightarrow{\theta}_d)$$

and $P(\overrightarrow{\Phi}|\overrightarrow{\beta})$
and $P(\overrightarrow{\Theta}|\overrightarrow{\alpha})$ are specified according to the modeling. For example, in the standard LDA settings where all terms are independent and exchangeable, we have:

$$P(\overrightarrow{w}_d|\overrightarrow{z}_d,\overrightarrow{\Phi}) = \prod_{i=1}^{n_d} P(w_{d,n}|z_{d,n},\overrightarrow{\Phi})$$
$$P(w|k,\overrightarrow{\Phi}) = \prod_{i=1}^{N} \varphi_{k,i}^{\delta_{i,w}}$$
$$P(\overrightarrow{z}_d|\overrightarrow{\theta}_d) = \prod_{i=1}^{n_d} P(z_{d,n}|\overrightarrow{\theta}_d)$$
$$P(z|\overrightarrow{\theta}_d) = \prod_{k=1}^{K} \vartheta_{d,k}^{\delta_{k,z}}$$
$$P(\overrightarrow{\Theta}|\overrightarrow{\alpha}) = \prod_{d=1}^{M} P(\overrightarrow{\theta}_d|\overrightarrow{\alpha})$$
$$P(\overrightarrow{\theta}_d|\overrightarrow{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \prod_{k=1}^{K} \vartheta_{d,k}^{\alpha_k-1}$$
$$P(\overrightarrow{\Phi}|\overrightarrow{\beta}) = \prod_{k=1}^{K} P(\overrightarrow{\varphi}_k|\overrightarrow{\beta}_k)$$
$$P(\overrightarrow{\varphi}_k|\overrightarrow{\beta}_k) = \frac{\Gamma(\sum_{i=1}^{N}\beta_{k,i})}{\prod_{i=1}^{N}\Gamma(\beta_{k,i})} \prod_{i=1}^{N} \varphi_{k,i}^{\beta_{k,i}-1}$$

Here, $\delta_{h,k}$ represents the Kronecker delta. Figure 4.1(a) graphically describes the generative process. As usual, the joint topic-data probability can be obtained by marginalizing over the $\overrightarrow{\Phi}$ and $\overrightarrow{\Theta}$ components:

$$P(\mathbf{W},\mathbf{Z}|\overrightarrow{\alpha},\overrightarrow{\beta}) = \int_{\overrightarrow{\Phi}}\int_{\overrightarrow{\Theta}} P(\mathbf{W}|\mathbf{Z},\overrightarrow{\Phi})P(\overrightarrow{\Phi}|\overrightarrow{\beta})P(\mathbf{Z}|\overrightarrow{\Theta}) \cdot P(\overrightarrow{\Theta}|\overrightarrow{\alpha})d\overrightarrow{\Theta}d\overrightarrow{\Phi}$$

In the following, we model further assumptions on both w_d and z_d, which explicitly deny the exchangeability assumption. Several other models can be obtained, which rely on more complex assumptions. However, the models delved in here subsume the main characteristics of sequential modeling. We observed that, in the real world, past decisions affect future decisions. In particular we focused on the behavior of a user base which is used to frequently buy items from a provider. A user tend to choose items according her tastes, but her tastes change over the time influenced by the purchased items. The sequence of these items depends on the fact that nearly purchased items are similar or share some features. For instance, let us consider the sequence of items $u.v.t$: initially the user bought the item u, then she chose v because of its similarity to u and finally she acquired t, that shares some features with v. Note that t should be completely different from u, but because of the taste change of the user they are in the same sequence. According to these assumptions, we choose to model the item sequence as a stationary Markov Chain of order 1:

- we choose to use a Markov Chain because of the sequential nature of the purchased item list, moreover the Markov Chain can model the user's taste changing over the time;
- the chain is stationary because users frequently buy items;
- the order of the chain is 1 because the probability that two subsequent purchases share some features or are dependent each other is higher than that of two purchases distant in time.

All these aspects lead us to the definition of the *Token-Bigram Model*, described as follows. We assume that $\overrightarrow{w}_d$ represents a first-order Markov chain, where, each item selection $w_{d,n}$ depends on the recent history $w_{d,n-1}$ of selections performed by the user. This is essentially the same model proposed in [79, 88], and the probability of a user trace can be expressed as

$$P(\overrightarrow{w}_d|\overrightarrow{z}_d,\overrightarrow{\Phi}) = \prod_{n=1}^{N_d} P(w_{d,n}|w_{d,n-1},z_{d,n},\overrightarrow{\Phi}) \tag{4.2}$$

In practice, an item $w_{d,n}$ is generated according to a multinomial distribution $\overrightarrow{\phi}_{z_{d,n},w_{d,n-1}}$ which depends on both the current topic $z_{d,n}$ and the previous items $w_{d,n-1}$. (Notice that when $n = 1$, the previous item is empty and the multinomial resolves to $\overrightarrow{\phi}_{z_{d,n}}$, representing the initial status of a Markov chain). As a consequence, the conjugate prior has to be redefined as:

$$\begin{aligned} P(\overrightarrow{\Phi}|\overrightarrow{\beta}) &= \prod_{k=1}^{K}\prod_{m=0}^{N} P(\overrightarrow{\varphi}_{k,m}|\overrightarrow{\beta}_{k,m}) \\ &= \prod_{k=1}^{K}\prod_{m=0}^{N} \frac{\Gamma(\sum_{n=1}^{N}\beta_{k,m,i})}{\prod_{n=1}^{N}\Gamma(\beta_{k,m,n})}\prod_{n=1}^{N}\varphi_{k,m,n}^{\beta_{k,m,n}-1} \end{aligned}$$

Since the Markovian process does not affect the topic sampling, both $P(\overrightarrow{z}_d|\overrightarrow{\theta}_d)$ and $P(\overrightarrow{\Theta}|\overrightarrow{\alpha})$ are defined as in equation 4.2. The generative model, depicted in Fig. 4.1(b), can be described as follows:

- For each user $d \in \{1,\ldots,M\}$ sample user community-mixture components $\overrightarrow{\theta}_d \sim$ *Dirichlet*$(\overrightarrow{\alpha})$ and sequence length $n_d \sim$ *Poisson*(ξ)
- For each user attitude $k \in 1,\ldots,K$ and item $v \in \{0,\ldots,N\}$
  - Sample item selection components $\overrightarrow{\varphi}_{k,v} \sim$ *Dirichlet*$(\overrightarrow{\beta}_{k,v.})$
- For each user $d \in \{1,\ldots,M\}$ and $n \in \{1,\ldots,n_d\}$
  - sample a user attitude $z_{d,n} \sim$ *Discrete*(ϑ_u)
  - sample an item $i_{d,n} \sim$ *Discrete*$(\overrightarrow{\varphi}_{z_{d,n},i_{d,n-1}})$

Notice that we explicitly assume the existence of a family $\{\overrightarrow{\beta}_{k,m}\}_{k=1,\ldots,K;m=0,\ldots,N}$ of Dirichlet coefficients. As shown in [88], different modeling strategies (e.g., shared priors $\beta_{(k,m),n} = \beta_n$) can affect the accuracy of the model.

By algebraic manipulations, we obtain the following joint item-topic distribution:

$$P(\mathbf{W},\mathbf{Z}|\alpha,\beta) = \left(\prod_{d=1}^{M} \frac{\Delta\left(\overrightarrow{n}_{d,(\cdot)} + \overrightarrow{\alpha}\right)}{\Delta(\overrightarrow{\alpha})}\right) \cdot \left(\prod_{k=1}^{K}\prod_{m=0}^{N} \frac{\Delta\left(\overrightarrow{n}^{k}_{(\cdot),m.} + \overrightarrow{\beta}_{k,m.}\right)}{\Delta(\overrightarrow{\beta}_{k,m.})}\right) \tag{4.3}$$

The latter allows us to define a collapsed Gibbs sampling procedure:

E step: iteratively sampling of topics, according to the probability

$$P(z_{d,n} = k|\overrightarrow{Z}_{-(d,n)}, \overrightarrow{W}) \propto \left(n^{k}_{d,(\cdot)} + \alpha_k - 1\right) \cdot \frac{n^{k}_{(\cdot),u.v} + \beta_{k,u.v} - 1}{\sum_{r=1}^{N} n^{k}_{(\cdot),u.r} + \beta_{k,u.r} - 1} \tag{4.4}$$

relative to the topic to associate with the n-th item of the d-th document, exhibiting $w_{d,n-1} = u$ and $w_{d,n} = v$.

M Step: estimating both $\overrightarrow{\Phi}$ and $\overrightarrow{\Theta}$, according to the following equations:

$$\begin{aligned} \vartheta_{d,k} &= \frac{n^{k}_{d,(\cdot)} + \alpha_k}{\sum_{k'=1}^{K} n^{k'}_{d,(\cdot)} + \alpha_{k'}} \\ \varphi_{k,r.s} &= \frac{n^{k}_{(\cdot),r.s} + \beta_{(k,r.s}}{\sum_{s' \in U} n^{k}_{(\cdot),r.s'} + \beta_{(k,r).s'}} \end{aligned} \tag{4.5}$$

Log-Likelihood.

The data likelihood, given the model parameters, $\overrightarrow{\Theta}, \overrightarrow{\Phi}$, is defined as follows:

$$P(\mathbf{W}|\overrightarrow{\Theta}, \overrightarrow{\Phi}) = \prod_{d=1}^{M} P(\overrightarrow{w}_d|\overrightarrow{\theta}_d, \overrightarrow{\Phi}) \tag{4.6}$$

Where:

$$\begin{aligned} P(\overrightarrow{w}_d|\overrightarrow{\theta}_d, \overrightarrow{\Phi}) &= P(w_{d,n_d}, \ldots, w_{d,1}|\overrightarrow{\theta}_d, \overrightarrow{\Phi}) \\ &= \sum_{k=1}^{K} P(w_{d,n_d}, \ldots, w_{d,1}|z_{d,n_d} = k, \overrightarrow{\theta}_d, \overrightarrow{\Phi}) \\ &\cdot P(z_{d,n_d} = k|\overrightarrow{\theta}_d) \\ &= \sum_{k=1}^{K} P(w_{d,n_d}|z_{d,n_d} = k, w_{d,n_d}, \overrightarrow{\Phi}) \\ &\cdot P(w_{d,n_d-1}, \ldots, w_{d,1}|\overrightarrow{\theta}_d, \overrightarrow{\Phi}) \\ &\cdot P(z_{d,n_d} = k|\overrightarrow{\theta}_d) \\ &= P(w_{d,n_d-1}, \ldots, w_{d,1}|\overrightarrow{\Theta}, \overrightarrow{\Phi}) \\ &\cdot \sum_{k=1}^{K} \varphi_{k,w_{d,n_d},w_{d,n_d-1}} \cdot \vartheta_{d,k} \end{aligned} \tag{4.7}$$

This formulation triggers a recursive procedure for the likelihood computation, whose trivial case is:

$$\begin{aligned} P(w_{d,1}|\overrightarrow{\Theta},\overrightarrow{\Phi}) &= \sum_{k=1}^{K} P(w_{d,1}|,z_{d,1}=k,\overrightarrow{\Phi})P(z_{d,1}=k|\overrightarrow{\theta_d}) \\ &= \sum_{k=1}^{K} \phi_{k,w_{d,1}},\cdot\vartheta_{d,k} \end{aligned} \tag{4.8}$$

Where $\overrightarrow{\phi}_k$ represents the initial state probabilities, as introduced above.

4.1.3 Item Ranking

The probabilistic framework is quite flexible, as it provides in general different choices for item ranking [75] an item for recommendation purposes. We next propose the functions relative to each model to be tested in the experimental section. In the following, we assume that a user can be denoted by a unique index u, and a previous history is given by $\overrightarrow{w}_u$ of size $n-1$. We are interested in providing a ranking for the n-th choice $w_{u,n}$.

LDA. Following [75] we adopt the following ranking function:

$$\begin{aligned} rank(i,u) &= P(w_{u,n}=i|\overrightarrow{w}_u) \\ &= \sum_{k=1}^{K} P(i|z_{u,n}=k)P(z_{u,n}=k|\overrightarrow{\theta}_u) \\ &= \sum_{k=1}^{K} \varphi_{k,i}\cdot\vartheta_{u,k} \end{aligned} \tag{4.9}$$

It has been shows that LDA, equipped with the above ranking function, significantly outperforms the most significant approaches to modeling user preferences. Hence, it is a natural baseline function upon which to measure the performance of the other approaches proposed in this part of work.

Token-Bigram Model. The dependency of the current selection from the previous history can be made explicit, thus yielding the following upgrade to the LDA ranking function:

$$\begin{aligned} rank(i,u) &= P(w_{u,n}=i|\overrightarrow{w}_u) \\ &= \sum_{k=1}^{K} P(i|z_{u,n}=k,\overrightarrow{w}_u)P(z_{u,n}=k|\overrightarrow{\theta}_u) \\ &= \sum_{k=1}^{K} P(i|z_{u,n}=k,w_{u,n-1})\vartheta_{u,k} \\ &= \sum_{k=1}^{K} \varphi_{k,j,i}\cdot\vartheta_{u,k} \end{aligned} \tag{4.10}$$

where $j=w_{u,n-1}$ is the last item selected by user u in her current history.

	IPTV1		IPTV2	
	Training	Test	Training	Test
Users	16,237	16,153	64,334	63,878
Items	759	731	2802	2777
Evaluations	314,042	78,557	1,224,790	306,271
Avg # evals (user)	19	5	19	5
Avg # evals (item)	414	107	437	110
Min # evals (user)	4	1	4	1
Min # evals (item)	5	1	5	1
Max # evals (user)	252	15	497	17
Max # evals (item)	2284	1527	9606	3167

Table 4.2: Summary of the evaluation data.

4.1.4 Experimental Evaluation

In this section we present an empirical evaluation of the proposed models which focuses on the recommendation problem. Given the past observed preferences of a users, the goal of a recommender systems (RS) is to provide her with personalized (and contextualized) recommendations about previously non-purchased items that meet her interest. Note that, although usually the standard benchmarks for evaluating recommendations are Movielens and Netflix data, they do not guarantee that the timestamp associated with each pair $\langle user, item \rangle$ corresponds to the timestamp of the effective purchase of the item, since the timestamp refers to the rating and the user may specify ratings in a different order. Moreover, we cannot rely on Videolectures data because, due to the privacy preserving constraints, this dataset do not provide user profiles but pooled statistics. We choose to evaluate the performances of the proposed techniques by measuring their predictive capabilities on two datasets, namely *Iptv1* and *Iptv2*. These data have been collected by analyzing the pay-per-view movies purchased by the users of two European IPTV providers over a period of several months [**?**, **?**]. The original data have been preprocessed by firstly removing users with less of 10 purchases the items with less then the same operation was performed over the items. We perform a chronological split of the data by including in the test set the last 20% purchases of each user. The main features of the datasets are summarized in Tab. 4.2. For each dataset, the users and items, in the test data, are subsets of the users and items within the training data. The sparseness factors of Iptv1 are 97.5% and 99.3% for the training and test sets (resp.), while the ones for Iptv2 are 99.3% and 99.8% (training and test sets, resp.). These values highlight the difficulty in discovering patterns and regularities within the data, in other words it's hard to define a good model for the recommendation. Fig. 4.2 and Fig. 4.3 show the distribution of the users and the bigrams (resp.) for both the datasets. As can be seen, these distributions exhibit the trend of power-laws [80].

Testing protocol.

Given an active user u and a context c_u currently under examination, the goal of a RS is to provide u with a recommendation list $\mathcal{R}$, picked from a list $\mathcal{C}$ of candidates, that are expected to be of interest to u. This clearly involves predicting the interest of u into an item according to c_u. We review here the evaluation metrics and the testing protocols to be used on this purpose.

In general, a recommendation list $\mathcal{R}$ can be generated as follows:

- Let $\mathcal{C}$ be a set of d candidate recommendations to arbitrary items;
- Associate each item $i \in \mathcal{C}$ with a score p^i_{u,c_u} representing u's interest into i in accordance to context c_u.
- Sort $\mathcal{C}$ in descending order of item scores p^i_{u,c_u};
- Add the first k items from $\mathcal{C}$ to $\mathcal{R}$ and return the latter to user u.

A common framework in the evaluation of the predictive capabilities of a RS algorithm is to split the traces $\mathbf{W}$ into two subsets $\mathbf{T}$ and $\mathbf{S}$, such that the former is used to train the RS, while the latter is used for validation purposes. Here, for a given user, c_u can be defined according to the technique under examination: the set of previously unseen items for the LDA, or the most recent preference for the Token-Bigram model.

In the latter model, it is required that all sequences in $\mathbf{T}$ precede those in $\mathbf{S}$, in order to provide a fair simulation of real-life scenarios. As a consequence, for a given user u, the trace $\mathbf{w}_u$ can be split into $\mathbf{w}_u^{(T)}$ and $\mathbf{w}_u^{(S)}$, representing the portions of the sequence belonging to $\mathbf{T}$ and $\mathbf{S}$, respectively. By selecting a user u, the set $\mathcal{C}$ of candidate recommendations are evaluated assuming $\mathbf{w}_u^{(T)}$ part of the context. The recommendation list $\mathcal{R}$ for u is then formed by following the foregoing generation process and the accuracy of $\mathcal{R}$ is ultimately assessed through a comparison with the items appearing in $\mathbf{w}_u^{(S)}$. Therein, the standard classification-based metrics, i.e., precision and recall, can be adopted to evaluate the recommendation accuracy of $\mathcal{R}$.

The latter can be defined according to an adaptation of the testing protocol defined in [81].

- For each user u and for each item $i \equiv w_{u,n}$ relative to a position n of $\mathbf{w}_u^{(T)}$:
  - Generate the candidate list $\mathcal{C}$ by randomly drawing from $I - \{i\}$.
  - Add i to $\mathcal{C}$.
  - Associate each item $j \in \mathcal{C}$ with the score $rank(i,u)$ and sort $\mathcal{C}$ in descending order of item scores.
  - Consider the position of the item i in the ordered list: if i belongs to the top-k items, there is a *hit*; otherwise, there is a *miss*.

By definition, recall for an item can be either 0 (in the case of a failure) or 1 (in the case case of a hit). Likewise, precision can be either 0 (in the case of a failure) or $\frac{1}{k}$ (in the case of a hit). The overall precision and recall are defined in [81] as the below averages:

$$Recall(k) = \frac{\#hits}{|\mathbf{T}|}$$
$$Precision(k) = \frac{\#hits}{k \cdot |\mathbf{T}|} = \frac{recall(k)}{k}$$

A key role in the process of generating accurate recommendation lists is played by the schemes with which to rank items candidate for recommendation. [75] provides a comparative analysis of three possible such schemes, and studies their impact in the accuracy of the recommendation list. It is worth noting that the score $rank(i,u)$ proposed here follows the main findings in that work.

Also, [75] shows that item selection plays the most important role in recommendation ranking. As a matter of fact, LDA turns out to be the model that best accommodates item selection in recommendation ranking, thus providing the best recommendation accuracy according to the above described protocol. It is natural hence to compare the Token-Bigram model proposed in this work with the LDA approach.

Implementation details.

All the considered model instances were run varying the number of topics within the range $[3,20]$. We perform 5000 Gibbs Sampling iterations, discarding the first 1000 (burn in period), and with a sample lag of 30.
Our implementations are based on asymmetric Dirichlet prior over the document-topic distributions (this modeling strategy has reported to achieve important advantages over the symmetric version [87]), while we employ a symmetric prior over the topic distributions. For the LDA and token-bigram models we adopted the procedure for updating the prior α as described in [85, 86]. We set the length of the candidate random list (see the testing protocol) equal to about the 35% of the dimension of the item sets for each test set. Precisely, these lists have 250 items for Iptv1 and 1000 items for Iptv2.

Results.

In Fig. 4.4 we summarize the best results in recommendation accuracy achieved by the proposed approach, over the two considered datasets. For each model, the number of topics which leads to the best results is given in brackets. On both datasets, the Token-Bigram models outperform the LDA models, both in recall and precision. At high level, these results suggest that exploiting the previous contextual information, the Token-Bigram Model outperforms LDA in recommendation accuracy. While the ranking function employed for LDA takes into account only the probability of selecting an item given the whole user purchase-history and the whole topic space, the Token-Bigram approach focuses on a region of the topic space determined by considering the previous item, thus providing a better estimate of the selection probabilities for the next user's choice.

In order to assess the stability of the proposed approaches in varying the number of topics, we plot in Fig. 4.5 and Fig. 4.6 the recall and the precision (respectively) achieved when the length of the recommendation list is 20. Considering Iptv1, best

results are achieved by both the techniques exploiting the largest number of topics we used for the experimentation, 30, with a recall of 0.347 and a precision of 0.017 for LDA and a recall of 0.379 and a precision of 0.019 for the Token Bigram, with a difference in recall of 0.032. For Iptv2, LDA achieves its maximum one again with 30 topics, while the proposed model has the best quality with 5 topics. LDA achieve a recall and a precision of 0.512 and 0.026 (resp.), while the Token Bigram has 0.556 for recall and 0.028 for precision, with a difference in recall of 0.044. It's interesting to note that the performances of the TokenBigram do not change substantially varying the number of topics. The results presented above experimentally prove the effectiveness of sequence-based topic models in modeling and predicting future users' choices. However those models increase significantly the number of parameters to be learned and this implies an increase in the learning time. In Fig. 4.7 we plot the learning time (5000 Gibbs Sampling iterations) for different numbers of topics. The learning time is consequently considerably larger. This is mainly due to the larger number of hyperparameters ($K \times K$ vs K) and to the complexity of the α-update iterative procedure.

4.1.5 Concluding remarks

In this part of work we proposed an extension of the LDA model. The proposed model relaxes the bag-of-words assumption of LDA, assuming that each token, not only depends on a number of latent factors, but also on the previous token. The set of dependencies has been modeled as a stationary Markov chain, which led us to define a procedure for estimating the model parameters, exploiting the Gibbs Sampling. This model better suites a framework for modeling context in a recommendation setting than LDA, since it takes into account the information about the token sequence. The experimental evaluation, over two real-world datasets expressing sequence information, shows that the proposed model outperforms LDA at the expense of an higher execution time when the number of the latent topics is large, since the number of parameters to estimate is bigger than in LDA.

For users of Caldera, the benefit of the recommendation systems are enormous. In fact, when surfing on the portions of the portal (subsets of graphs made by *Borè*) they are suggested to other sections (other subsets of graphs).

4.2 Collaborative Editing

In recent years, organizations are relying more and more to the Internet for the management and publication of data. This offers important opportunities for collaboration with users of these organizations, mainly in the writing of documents. From this collaboration comes the problem of concurrency control of shared documents for authoring software. The most common techniques to manage concurrent access to shared documents can be categorized as follows:

- *Locking*: the document is accessed for writing by only one user at a time.

- *Versioning and conflict resolution*: you can store the revisions of a document, and in the case of conflict resolution is delegated to the end user.
- *Real-time collaborative editing*: multiple users access to write to the same instance on the network of the document, watching the changes of others and resolving conflicts in real time.

Over time, users have had access to greater computing power and lower latency in the network. This allowed the paradigm of real-time editing of being the center of groupware systems.

Applications that allow multiple users to work on a common shared object, using as a means of a computer network, are called group editors.

There are many group editor currently on the market such as: Google Drive, ACE, EtherPad, Gobby, AbiWord, Mozilla Skywriter and SubEthaEdit. Some of these are open source, such as ACE, EtherPad, AbiWord, Gobby and Mozilla Skywriter, of others, however, do not know the technology. Many of the group editor on the market are desktop applications, developed before the Web 2.0 phenomenon. New applications are Web and are based on technologies AJAX and Comet, to be immediately used by Internet-users.

Research in this field has produced several collaboration systems, many of which are text editors, such as GROVE [45], REDUCE [46] and Jupiter [47].

The same principles of the related algorithms, however, can be applied to more complex data structures than a simple linear text. The group editors as well as being collaborative tools represent a technical challenge and an incentive to the study of problems of consistency. In fact, unlike to the traditional systems of control of the consistency in distributed systems, the group editor take into account the human factor.

The traditional techniques used by the group editor are locking, turn taking, serialization and causal ordering [48, 49]

The most modern and innovative technique is called Operational Transformation (OT). It is used by the current group editor. There are various systems based sull'Operational Transformation, each of which solves the problem of consistency with algorithms and different consistency models. This approach has been studied for many years and still today we seek solutions in the literature increasingly efficient and innovative.

4.2.1 Goal

The purpose of this part of the thesis is to introduce the problem of simultaneous editing within our platform shared documents and study techniques aimed at the Operational Transformation that solve this problem and keep the user interface intuitive and accessible at the same time .

This work stems from the argument [29]. I was the rapporteur. The work is much broader and more complex than shown here.

After the theoretical study of the techniques OT has worked on implementing two algorithms, Jupiter [47] and SOCT3 [55] , In the Web 2.0 [56] and Web 3.0 for collaborative real-time editing of HTML FORM. Of such algorithms will be also carried

out the testing. was chosen as the algorithm SOCT3 suitable choice for integration in Caldera (and even before in Bor)

Note that all algorithms alvorano on a single text field, a document, a single form. One aim of the work was the integration is therefore to support the simultaneous editing of documents comprising multiple fields primitive or composite by multiple users of the platform.

4.2.2 Benefits for end users

The technique Operational Transformation, which has been studied and materialized uses a very different approach from traditional ones: closer to the user and the collaboration process is more natural.

Traditional methods such as locking and turn taking delay changes from the user, making them slow and requiring the user to perform tasks not relating to the writing of the document.

Other methods, such as version control systems, resolve conflicts by working on the final document. When the user saves, does not take into account the changes of other users, creating potential conflicts when saving. In case of conflict, the machine does not have enough information. In fact, does not know the intentions of the user. The resolution is delegated to the end user.

With the method "Operational Transformation" we can edit a document simultaneously shared by multiple users, each user has a local copy of that document. For each user action immediately is the generation of an operation that encodes the intention of such action, this is transmitted to all other users.

The remote operation is received from the application of each user, it is transformed and subsequently applied to the local copy, so that the user can take account of the changes that have been made to the document at any time.

At this point the problems of traditional methods are solved and there is no conflict at any time during editing session. [57]

In fact, every member of your editing session will modify the document having knowledge of the changes made by other users in real time. The user does not need to wait at any time to make changes, it has to deal with conflict resolution and transactions made by other users are transparently applied during the process of collaboration.

4.2.3 Contribution of Working

The work summarizes the steps made by research in the Operational Transformation, the various models of consistency and what has been achieved to date. Has been studied how this technique can be implemented with the technology of Web 3.0 and how it can be applied to individual fields in a FORM. A type of application of real-time collaborative editing as yet unseen in the world Web Implementing SOCT3 it is highlighted and analyzed a property of the algorithm, supported by theoretical demonstration, which allows to obtain significant simplification in terms of complexity of the code. After a thorough analysis of the implemented systems have been

described, the reasons for the choice of the algorithm SOCT3 rather than Jupiter. It was created for the client-server paradigm and it is adopted by most of the group editor on the market. The integration of the component SOCT3 Caldera constitutes an innovative change for the same platform

4.2.4 The problem of simultaneous editing

For real-time collaborative editing refers to the technology that allows multiple users via an editor and a computer network, to be able to work simultaneously on a shared document.

The software offers the user the possibility of this collaboration is called collaborative real-time editor. The concept of collaborative real-time editing with Douglas Englebart dates back to 1968, but it took many years before getting the first implementations.

The technical challenge is to find the most natural to the user's eyes to apply the changes made by other users. There are different techniques to achieve the desired effect. Many of these such as locking and turn taking slow user actions.

Others such as serialization, causal ordering and transformation can not preserve the intention that the user had at the time of the change.

The main problem is that, due to the latency of the interconnection network between the users, the propagation of changes is delayed. It follows that a user may have performed operations on a document without taking into account the changes made by other users.

The technique proposed in this work in order to solve this complex problem is the Operational Transformation.

4.2.5 Introduction to Operational Transformation

The Operational Transformation (OT) is a technology to support group editing systems. This technology was invented, and over time widely accepted by the groupware systems, to manage the consistency and competition in the writing of various documents (texts, structured documents, media design, etc..).

At the base of of the concept of OT there are *users*, a *shared document* and the *operations* that can be performed by users of this document. Each time a user in a session of group editing changes to the document, this change is propagated to all other users of the session. In this way, each user has knowledge of the changes made by other users in real time, and the final document is to be consistent.

The OT systems in general are able to offer several features:

- managing the consistency and competition;
- conflict Resolution;
- group undo;
- locking
- ompression of operations;

To give an idea of the systems OT we can think of three sites in a session of group editing of a shared document, as shown in Fig. 4.8

Despite the exchange of messages represents a use case text editing linear daily very simple, it is complex because of the network latency. The generation of some operations may in fact occur in any instant because there is no locking in the UI of the user, as well as the reception of messages and their application.

In this particular example, it is expected that the final content of the document is ”Hey Pippo”.

n fact, if we analyze the performance of the operations:

1. User 1 and 3 concurrently insert ”Hello” and ”Hey.”
2. User 2 receives the message from user 1, but not yet by the user 3 and erase the whole word written by the first user. The status of the document to the user's eyes 1 turns out to be ”Hello”, that user 2 and user 3 and void and that ”hey”.
3. User 1 enters ”Pippo” finally getting the text ”Hello Foo”.
4. User 1 User 2 receives the command getting ”Foo”, after which receives the user's operation 3 getting ”Hey Pippo”.
5. User 2 receives the operation by the user getting 3 ”Hey Pippo
6. User 3 receives the first operation 1 user getting ”CiaoHey”, after which the user 2 to give back, ”Hey” and finally ”Hey Pippo”.

In the last point we can highlight the fundamental problems of real-time collaboration: Do users enter the text in the same position in which order will be applied to the operations? If the order had been ”HeyCiao” as does it would apply the delete operation? The solution is to perform the transformations of these operations to allow all clients connected to converge to the same state of the document. The innovative idea of the OT, compared to traditional methods, is to transform the operations in reception compared to those performed concurrently

4.2.6 The model

To overcome the complexity of the problem it is necessary to first define the terms and theoretical concepts that form the basis of this technology.

Definition 4.1. *An object is an entity with which the user can interact in order to edit a document.*

For example a text area where the user can insert or delete text. For simplicity we will refer to a single object of type plain text WLOG

Definition 4.2. *A transaction is the translation of an event generated by the object in a certain formalism specific application.*

If an area of text the user enters the word ”bob” in position 10 (assuming that the location is valid, that is, that there are at least 10 characters), then the operation will be generated $ins, "bob", 10$ If the user deletes 3 characters in position 5, then the operation will be generated $del, 3, 5$ Each object will have its own set of operations that is application-specific and algorithm.

Definition 4.3. *A site is a group which participates in the editing system connected to a communication network.*

Assume that architecture is client-server where a site corresponds to the generic client connected to the server. A user may open multiple sessions with the groupware system and therefore more sites. For simplicity, a site will be uniquely identified by integer WLOG

Definition 4.4. *An editing session is quiescent in a given instant of time, if all the operations generated were performed on all the sites [45].*

That is, the system is quiescent when there are more requests waiting to be processed, or in a way the users are in the "break."

Definition 4.5. *The state of an object is the representation of the contents of that object in a certain instant of time.*

The execution state of an operation is the status on which this operation will be applied. For a text document, the state could be represented by the text itself.

Definition 4.6. *The intention of a transaction is the effect that the operation at the time when it was generated by the user [49].*

This definition is in fact very general and is therefore very specific application domain. As an example we can assume to have an operation of , 1 generated in the state *abc* , Which leads to the state *c*. The intention of the operation is to delete the letter *b*

Definition 4.7. *Given two operations O_i and O_j generated respectively to the sites i and j , Is said to O_i causally precedes O_j*
$O_i \longrightarrow O_j$ *iff [49]*

1. $i = j$ and the generation of O_i is happened before of O_j, or
2. $i \neq j$ is the execution of O_i on site j
   is happened before of the generation of O_j
3. exists an operation O_k such that $O_i \longrightarrow O_k$ and $O_k \longrightarrow O_j$.

This relation is transitive, antireflexive, antisymmetric. In particular, the transitivity is to be managed by the management algorithms of competition.

Definition 4.8. *Given two operations O_i and O_j it is said that O_j depends O_i iff $O_i \longrightarrow O_j$ [49]. Viceversa O_i and O_j are indipendent, or concurrent ($O_i \| O_j$) iff $O_i \nrightarrow O_j$ and $O_j \nrightarrow O_i$.*

There are various models of consistency of OT systems, each of which provides different properties in terms of correctness of the algorithm:

- *Causalty precedence*: given two operations O_i and O_j such that $O_i \longrightarrow O_j$ then on each site execution O_i must take place before the execution of O_j.

- *convergence*: ensure that the replicated copies of the document are identical for each site when the session is quiescent.
- *Intention preservation*: ensures that the effect of the execution of an operation on a generic status of a document reflects the intention of the operation (ie, the effect of executing the operation at the time of its generation). This property has been formalized only in new theoretical frameworks [58–60] .
- *Single-operation effects*: the effect of the execution of a certain task in a certain state has the same effect as what you would have in the state in which it was created.
- *Multi-operation effects*: the report to the effect achieved by the execution of any two operations remains even after their execution in any state.
- *Admissibility*: the invocation of each operation is permissible in its execution state.

The consistency models emerged from the study of this technology are:

- *Causalty, Convergence (CC)*: this was one of the first models which can not ensure convergence in special cases of concurrence.
- *Causalty, Convergence, Intention Preservation (CCI)*: this model guarantees the convergence and is a generic model and model-independent data.
- *Causalty, Single-operation effects, Multi-operation effects (CSM)*: This model seeks to formalize the concept of Intention Preservation.
- *Causalty, Admissibility (CA)* very recent model that seeks to formalize the concept of Intention Preservation, currently applicable only on linear text documents.

We have focused on the study and implementation of algorithms and CCI of the theoretical basis of the CA model. In fact, the CCI model is most commonly used in both desktop and web applications and is the most studied and currently valid.

4.2.7 Algorithms

4.2.8 Algorithm dOPT

The algorithm used in Dopt GROVE cite elis was the first to address the issues of the Operational Transformation in a CCI model. At this stage, the theoretical basis was not yet complete, in fact they had not yet formalized the properties TP1 and TP2, and the concepts of context-equivalence and context-precedence.

The main features are:

- A log linear structure called *log* where each site maintains its history of operations performed.
- If an operation O receipt is not causally ready, it is placed in queue until it is causally ready.
- A vector called state vector SV_j for each site j of the form $[e_1, e_2, ..., e_n]$, where $\forall i = 1, ..., n$ e_i is the number of transactions generated by site i that have been performed on site j.

- each operation O_i generated at a certain site i has assigned the timestamp $O_i.SV_itimestamp = i$.

Theorem 4.9. *an operation O_j generated at site j with timestamp $O_j.timestampis$ is causally ready on a site i with state vector SV_i and $i \neq j$ iff the following conditions are met [49].*

1. $SV_i[j] = O_j.timestamp[j] - 1$, that is i has completed the operations of the site j except O_j.
2. $SV_i[k] \geq O_j.timestampe[k] \quad \forall k \neq j$.

The first condition ensures that the website i has completed the operations of the site j except O_j.

The second condition ensures that the website $j$has performed all the operations that causally precede O_j, then is satisfied the property of Causalty Preservation.

Despite dopt able to solve many problems of competition, however, is to fail in a use case called dopt Puzzle [49] , showed in Fig

4.10 version of an example.

In this scenario we have

$(O_1 \longrightarrow O_3) \| O_2$

On the site 1 4.11(a) we have O_2 *sqcup*O_1 and consequently $IT(O_1, O_2)$ *sqcup*O_3. The result O_2^{prime} is therefore correct. On the site S_2 4.11(b) instead we find ourselves in the next phase of execution $O_1^{prime} = IT(O_2O_1)$.

In this case O_3

is not context-equivalent with O_2 , by obtaining a O_3'

that violates the convergence properties.

In the work of Dopt is not proposed a system of garbage collecting very efficient, as it empties the log periodically forcing the quiescence of the session. This means to grow the log and at a certain moment ”pause” users, without which they can edit the document.

4.2.9 Algorithm GOT

The algorithm Generic Operational Transformation (GOT) used in REDUCE [49,52]

Exclusion Transformation introduces the functions and the concept of the execution context of a transaction. How dopt, also uses the state vector as a technique for timestamping and a one-dimensional buffer for the operation history called $HISTORYBUFFER$ See [29] for more details.

4.2.10 Algorithm GOTO

The inefficiency of the algorithm GOT due schema undo/do/redo has led to the study of a new algorithm GOT Optimized (GOTO) always REDUCE [51].

This new algorithm no longer provides TP2 property of the control algorithm but in the transformation functions. For this reason, the operations of undo/redo are no longer necessary.

The algorithm is in fact very similar to GOT, except this simplification in the control part.

The problem in this case, as previously discussed, is in the complexity of the transformation functions. While the algorithm, in fact very similar to GOT, simplifies the control part, the inefficiency and complexity moving towards the transformation functions, not improving the situation in general. The transformation functions require additional parameters in order to satisfy the TP2 and must consider many more cases. See [29] for more details.

4.2.11 Algorithm Jupiter

The algorithm Jupiter [47] comes to simplify the control algorithm in the case of architectures client-server. Key features of this system:

- There is a central server which performs most of the functions.
- Clients and servers implement algorithms, even if only slightly, different.
- Use a two-dimensional state space.
- It has queues for messages received.
- It has a queue of outgoing messages called on Outgoing.

The algorithm assumes that there is a pair-wise communication between client and server, and you may receive the following assumption:

Definition 4.10. *For each pair of messages sent by the server in chronological ordere O_i and O_j , it has $O_i \longrightarrow O_j$ $O_i \mapsto O_j$.*

Each state is represented by two numbers $(myMsgs, otherMsgs)$, where $myMsgs$ is the number of operations of the client and $otherMsgs$ is the number of server operations that are processed locally by the client. Unlike previous algorithms that use it as a timestamp of the state vector with number of elements equal to the number of connected sites, Jupiter in the timestamp is only the pair $(myMsgs, otherMsgs)$. Technically, the algorithm performs the operations of Jupiter merging the server side, so it must maintain the state space of each client and all information necessary for its proper operation.

4.2.12 Algorithm SOCT3

The algorithm Serialized Operational Transformation 3 [55] has similar characteristics to other algorithms: using a one-dimensional buffer named $History$ and uses the state vector to determine the causality of operations. SOCT3 solves the problem undo/do/redo GOT without satisfying the property TP2 in the transformation functions as GOTO. To do this we introduce the concept of Continuous Global Ordering (CGO).

Definition 4.11. *The continuous global order is an order, recognized globally in the system. The continuity of the order can not be violated.*

In the simplest case, the order of the natural numbers 1, 2, 3, ... is a CGO. With this order we can not have, for example 1, 2, 5, 8, as has discontinuities, or 1, 2, 2, 3 because it contains duplicates.

For each operation generated is assigned an ID that is an element of CGO. In our case we choose W.L.O.G. natural numbers as CGO. In fact, this ID uniquely identifies the transaction in the system, as opposed to the timestamp might look the same for most operations.

The identifier, to be such, must be unique for a given object within the system, ie at any instant of time$\nexists O_i, O_j$ such that $O_i.id = O_j.id$. In the theoretical treatment, the generation of this ID is delegated to a function $ticket()$ for a specific application domain, which returns every call a monotonically increasing integer value obtained from a *sequencer* [**?**] , and that, overall, has no discontinuity.

Preposition: Given two time instants $t_1 < t_2$ and given $id_1 = ticket()$ at time t_1 and $id_2 = ticket()$ at time t_2 , then it must be true $id_1 < id_2$.

Theorem: Given two operations O_i and O_j such that $O_i \longrightarrow O_j$ then $O_i.id < O_j.id$.

Proof if O_j follows causally O_i means that at the time of generation of O_j the operation O_i had already been performed locally, then the call to $ticket()$ for the allocation of $O_j.id$ was made certainly in the next instant to that for $O_i.id$. As a consequence $O_i.id < O_j.id$

Unlike Jupiter algorithm, the algorithm SOCT3 is not meant for a specific client-server or P2P systems [61] Although the theoretical treatment refers to a model where each client with a broadcast communicates directly with every other client, and there is no central entity that maintains a copy of the document.

This feature can be used to lighten up the load on the server and load more clients. While Google Docs lightens up the server delaying the propagation of changes and maintaining a single state space, in this work is not handled any state space and the propagation of changes is immediate. The server does not perform any transformation, and therefore does not know the specific format of the messages sent between the client, causing it to remain unchanged with respect to any future extension of the client side.

Most of the instructions in the implementation refer to a particular widget, which will be often omitted to simplify the exposure. State of the algorithm for each client-side widget (in brackets the initial values of the variables):

- *lastId* (0): ID of the last operation delivered
- *genId* (-1): next value of the local ID to be assigned to the transactions generated.
- *lastRecvId* (0): lastID operation received from the server.
- *lastAutoSavedId* (0): lastID of the last auto-save.
- *lastAssignedId* (0): lastID assigned to the operations sent from this client.
- *waiting* (false):flag indicating if the algorithm must not perform the delivering of operations.

4.2.13 Algorithm Admissibility-Based

This new technique Admissibility-Based [60]

, which reflects a model of type CA, is very recent (2010, Nokia and Google, used on some mobile phones), and it is very different from the techniques seen so far. The main features that differentiate this method from others are:

The formalization of the concept of Intention Preservation, specifically for linear texts.

- Storing erased characters and the story of these operations, in order to transform the operations ensuring Intention Preservation with rigor.
- Separation of the history of operations according to the type of operation, in the case of linear texts you have left erase operations and right insertion operations.
- There is no separation between the control algorithm in the architecture and transformation functions.
- view the text document as a linear graph, where nodes are the characters and the arcs correspond to the sequence of characters in the text.

Also here you can prevent the TP2 property is satisfied by the transformation functions

The work is formally valid and an interesting theoretical basis for future work, mainly with the aim to extend the technique to support more complex types of document. Articles have been published that extend the set of operations, but also for linear text documents.

4.2.14 Google Wave

Google has adopted the Operational Transformation system for Google Wave. [62] Wave was closed but then the technologies have been used for Google Docs. It is known that the algorithm used is derived from Jupiter, but do not know the implementation details.

Jupiter heavily loaded the server because you have an instance of the algorithm for each connected client. From the published documents, Google has found a solution for which the server maintains a single state space regardless of the number of users, greatly easing the computational load on the server. To implement this solution, it slows down the propagation of changes made by the user on the document, keeping in cache operations generated locally that will be sent to the server only when you reach certain conditions.

Despite this, the algorithm tries to guarantee a delay rather limited, also thanks to the computing power and network bandwidth.

4.2.15 Integration in Caldera

Caldera (and before Bor [63]) Brings the user-experience in the web pages of the portal through AJAX technologies and the Dojo framework.

Each page in Caldera is composed of fragments that are loaded dynamically, without changing the browser page, by giving the user more efficiently. Clicking on a link generally is updated on the main fragment, without reloading the rest of the

fragments. This means that the system initialization seen previously can not rely only events onload and onunload of the HTML page.

The part that competes algorithm OT is the editing. The editing form is loaded in the main fragment called contentPane, built with the component that provides Dojo.ContentPane events onload and onunload.

At this point we are faced with one crucial difference: whereas before the change of a page re-initializing the entire namespace JavaScript, JavaScript objects are now shared during navigation application Caldera. The consequence is that Site will no longer be a singleton on the page, but every time you enter the editing phase we need to create a new instance, getting the new life cycle of the widgets shown in Fig. 4.12.

If before it had only one instance of Site, now every time the user enters the editing mode you create a new instance, and you bind the widgets in that instance. When the user exits the editing session, the site is disconnected and is made unbind widgets.

The unbind the widget is primarily to stop the various timers, disconnect events and block the algorithm because the environment has changed. Actually unregister the message is sent by a server-side form of APE, written specially when the site is disconnected. This choice is dictated by the fact that the Comet server automatically handles the timeout of the client.

4.2.16 Extension of the algorithm

Documents in Bor contain several widgets that so far we have not covered. In a sense, we extend the algorithm to support new widgets and new business. Below is a list of the supported widget:

- Text fields (string, rawText).
- Numeric fields (int, double).
- Date fields, or date and time (date, datetime).
- Drop down menu
- Dynamic list of widgets (for the cardinality).

The numeric fields are represented with the special input of type text in the HTML form, so as text fields, but with a constraint: the content must be a valid number. This constraint is enforced by the framework Dojo. For these widgets numerical IntWidget and DoubleWidget, you chose to inherit from LastValueWidget.

For a list of the widget instead, ListWidget, the situation is more complex.

Dynamic list of widgets

This widget is composed of sub-widgets. The type of sub-widget is always the same and is passed to the constructor of the class ListWidget (eg TextWidget, IntWidget, etc..). In the method $ListWidget.doConnect()$ we make use of the static methods $findElements()$ and $getFieldName()$

to build each sub-widgets already present at the time of loading of the form editing.

Each sub-field is uniquely identified in the list with an item number, generated with a monotonically increasing function.

In terms of UI, the user can perform three fundamental actions:

- Add a new field to the list.
- Delete an existing field from the list.
- Modify an existing field in the list.

While the third action is managed directly by sub-widget, since it directly captures the events of the browser, the first two are the responsibility of ListWidget. The operations are of the type

$(add, ordinal, count)$ and $(del, ordinal, count)$ where count

is the number of sub-fields present at the time of generation of the operation.

In the definition of the operations must take into account that the lists are subject to a cardinality constraint. In Caldera you can have cardinality of the type

$a : b$ with $a \in [0, n]$ and $b \in [1, \infty]$, where $0 < n < \infty$, ie there may be lower and upper limits. When it exceeds a lower limit a warning is displayed in the UI, then the algorithm can in practice ignore the fact that there is a lower limit.

The particular cases that have emerged are:

1. Suppose that a list L with cardinality $0 : n$
   has two fields Two users simultaneously add a new field, generating the operation $(add, 3, 2)$. At the merge is expected on both sites are added two new fields with ordinal respectively 4 and 5.
2. Suppose that a list L with cardinality $0 : 4$ has three fields. Two users simultaneously add a new field, generating the operation $(add, 4, 3)$.
   To merge it is expected that the number of fields is 4 and not 5 as in the previous case. In fact, the intention of the individual user was to add the fourth field, the last, since the user is aware of cardinality.
3. Suppose we have a list L with 3 fields respectively with items 1, 2 and 3. Two users simultaneously erasing the second field generating operation $(del, 2, 3)$. It is expected that on both sites is removed exclusively the second field.

To meet the first case it introduces a third operation $(insert, ordinal, count)$. The effect of this operation is to insert immediately before the field with the ordinal *ordinal* a new field, increasing the ordinal of all subsequent fields.

For the second and third case, we introduce another operation(nop) that has no effect at the time of its application.

Generation operations from under-widget

The generation of an operation by a sub-widget does not take account of the fact that it is contained in a ListWidget. For example, a text field may generate $(replace, 1, 3, abc)$. As we have seen in $apply()$

it is necessary to know the attribute ordinal in some sub-widget apply the operation. To this end, a function is created $transform()$, to be passed upon initialization with $subwidget.connect()$

, which is called by $emitOp()$ every time that the sub-widget generates a new operation.

this function $transform(subwidget, op)$ for ListWidget does is assign:

$$op.ordinal := getOrdinalOfSubwidget\,(subwidget)$$

$$op.count := getCurrentNumberOfSubwidgets()$$

4.2.17 Saving the document

In order to implement the algorithm is modified database Caldera. The database will contain Caldera, therefore, the tables with the contents of each document when it is saved by the user, and tables specific to the OT.

In the platform Caldera documents are saved in the database of user action, and there is a versioning system to resolve conflicts in the case of simultaneous backups. The model of consistency implemented in this thesis allows users to edit a document to have the local copy of the updated form, so that the rescue by a user also includes the modification of other users.

It is precisely for this reason that you can avoid checking for conflicts in the versioning system.

Furthermore, the server can reach more saves by users. If a saving is old, that does not account for multiple operations of other users, then it must be ignored. The term for this condition is to verify the *lastId* of the site making the rescue, just as for self-rescue.

Saving can be for the algorithm OT another time to clean the database, how you do it for self-saving. This is exactly the same algorithm is proposed for self-saving for the saving.

The only difference lies in the message that the client sends to the server:

- Saving sends the data for all the widgets, while the auto-save is done for a single widget.
- The message for the algorithm OT does not include the content of individual widget.

The content of individual widgets is no longer necessary for the algorithm OT because the document is saved directly in the tables of documents Caldera.

4.2.18 Summary of an editing session

An editing session from the point of view of saves and self-saves can be represented with the statechart in Fig. 4.13

When there are no active sessions on editing a document, the first user receives the data saved in the model. By changing the document you generate more transactions until it reaches the *MAXBACKLOG* . At this point snaps to the self-saves and tables algorithm OT field values are saved in the document, those values that are displayed in the user interface, whether they are valid or not. And so on for further self-rescues.

When you save the values of these fields can be reset in the tables of the algorithm OT. In fact, if a new site is added to the editing session, the HTML page will be loaded with data from the model of Caldera, avoiding an unnecessary copy in the algorithm.

4.2.19 Concluding remarks

Initially we have introduced the technique Operational Transformation and how it can innovate the processes of collaboration between users. OT techniques have many advantages, such as being able to preserve the intention of the users in carrying out certain actions, but they can not represent the exact semantics of such shares at the time of the resolution of a conflict.

We did a study of the various algorithms and have proven to be proficient in supporting the group editing systems, replacing traditional techniques. Of each of these algorithms, we have highlighted the advantages and disadvantages, and the substantial differences that exist between the various approaches.

We have designed and implemented an extensible framework for client-server architectures, analyzing aspects of both client-side server side, and how the various components, while working synergistically, are separated and therefore replaceable (decoupling & code reuse). This framework is applicable for group editing in any context of Web 2.0 and Web 3.0, on any type of object that has interaction with the end user.

The control component of this framework, which manages how and when to perform remote operations, we have implemented the consistency models Jupiter [47] and SOCT3 [55].

In particular, it was concluded that SOCT3 is a good solution for client-server architectures.

We have formalized a specific property SOCT3 algorithm that greatly simplifies the writing of the transformation functions, ie those functions that transform operations for the resolution of conflicts.

Finally, we have integrated the thesis work in the Document Management System Caldera in order to support collaborative real-time editing of complex documents by the users of the portal. In this phase has been solved the problem of saving centralized shared document as well as integration with the navigation system Web 3.0 of Caldera.

HTML objects currently supported by the system are linear texts, dropdown menus, dynamic lists of sub-objects or other objects easier.

Research and some of the applications of group editing, as EtherPad and Google Docs, have gone further by allowing collaborative editing of structured content [64] with editor rich text [54] and markup languages [65] , and even graphic editors [50].

It is clear that the work has to do with recent technologies and which continue to be studied in research. In fact, the release of Google Wave [62] in 2009 sparked further interest in the world of real-time group editing, as well as the new framework [53,66] and the same algorithm Admissibility-Based [60] of 2010, which proposes a more formal approach than traditional systems.

4.3 Caldera architecture scheme

Finally, after this extensive discussion, the work of study of the platform has come to an end. Of course there is still much to do and it much will be done. In Figure 4.14 you can find the final structure of our platform.

Every single component has already been analyzed, therefore we do not write more. However, in the next section we will discuss a use case of the platform.

4.4 Use case: Condomani.it

In this section we analyze a vertical solution of Caldera platform which is currently available for use by real users. the platform is owned by Condomani Srls, a company of which I am CEO & founder. In fact, during the doctoral studies, I tried to apply the concepts in a practical way. From this was born the project that is enjoying considerable success. is worth pointing out that although the guidelines of the construction of the platform are those of the Caldera, the code used is different because the project is much larger and must be vertical system. But the first version of Condomani [3], was achieved through Caldera. To date in Condomani representatives are many of the concepts of Caldera, and many others, will soon be implemented.

The reason why we describe below a commercial product is not to get publicity. The reason is that we want to show how this argument representing the frontier of research began to become a reality right now.

4.4.1 The idea

Condomani is the social network for the management of the condominium. It is aimed to three kind of users: Flat Owners, Condos Administrator, Service Provider. The mission is make life easier in the condominium. With Condomani the whole building is to be finally made active part.

Research side. To realize it was based on the concepts of Social and sharing informations studied and described above.

4.4.2 Users, product and service description

Condomani is aimed to three kind of users: Flat Owners, Condos Administrator and Service Provider.

Just analyzing the example of the Italian market we discover large numbers. There are 1 million condominiums, managed by 320,000 administrators with a total expenditure of 19 billion.

Condomani combines the functionality of the management with the features and scalability of social networks for communities, qualifying it as a first mover into a space that is not currently controlled.

[3] http://www.condomani.it/

- The condos administrator are the target that buys the licenses.
- The flat owners are the target that populates the platform and summon service providers
- The interaction between administrators, owners and providers is the lever to scale the platform.

Condomani is the social network for the management of the condominium. It is available at the link www.condomani.it and is fully functioning. It is web based. With Condomani a flat owner can submit a request for action to its Condos Administrator, even from his smartphone; monitor the progress of any work of condominium; check all the expenses; dialogue with other flat owners. Condomani also supports throughout the Condos Administrator: from condominium meetings to budgets, from requests for assistance to requests for quotes. The administrator can receive reports made by condominiums and forward requests for emergency providers default in just one click. In addition, he have the ability to create an auction transparently between the different service providers to request quotes. The administrator can forward each type of alert (e-mail, sms, fax, mail, voice messages) to all its condominiums, even those who do not use Internet, sitting comfortably on his couch. The latest figures that come into play are the services provider. With Condomani they will be able to: receive requests for assistance; update on the evolution of the work; find new customers by launching offers; receive feedback on their operated. These are just some of the many functions of Condomani that are revolutionizing the communal life of the people who have already chosen to use it. In Italy the reform of Condos was approved some months ago. Among the keywords we find website condominium and transparent budget. This is exceptional for Condomani.

Research side. As can be seen, there are three types of user. Each type of user has its own universe and various documents and connected informations. For fast prototyping work has been done on the nodes and the edge of our platform. They were well-defined groups and the types of person. Next, we defined the workflow and the correct questions to have all the necessary information.

4.4.3 Problem solved and innovative elements

What will happen to apartment buildings? Who is our manager? Who is the best gardener in the area? These and a thousand other questions remain open. The management arrangements of the building are old and not in line with the new social and technological trends. The flat owners often do not know the real work done by his conds administrator, as always absent in times of need and not very clear when it comes to motivating costs and outputs. On the other hand, administrators do not always have the necessary tools to make the various cooperative and collaborative condominiums. There are Facebook and smartphone, but the condos are always the same: the technology and benefits of it does not have minimally affected the way people live their lives condominium. With Condomani all this questions can be solved in an efficient way, and all of us can save money and time.

Condomani is an innovative product because it optimizes and innovates a classic market and obsolete: the world of the condominium. There are investigations,

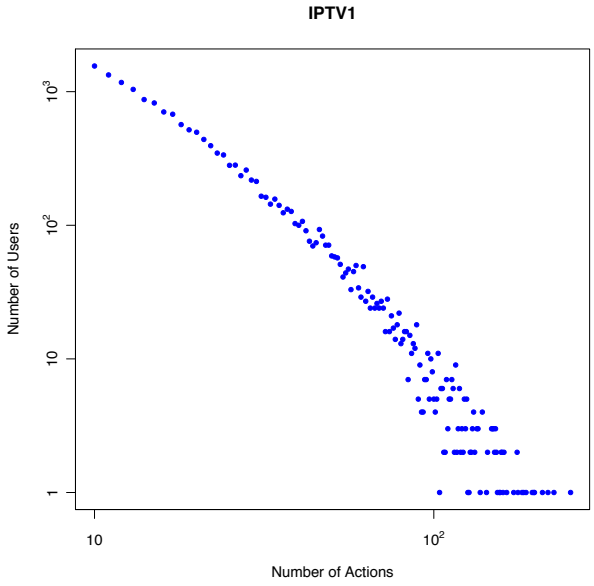
surveys, case studies, idioms, etc.., that say the building administrator scam and the neighbor is a noisy and unpleasant person. To date, there is no product that allows a true innovation in the building. It seems that the social networks and SaaS products does not touch the condos. Here is our innovation: bringing in condominium a social network management. We have already done. It is SaaS and Cloud. Finally, the condominium is in your hands. These are the main innovations of Condomani that allow us to position ourselves well above our competitors, to reduce costs and waiting times for customers, allowing us to have a high quality product but lowering costs. What else creates our innovation? It seems strange, but it creates happiness among users.

Research side. We have exploited the mechanisms of Recommandation and Process Mining to develop valuable information to provide the team with analysis and sometimes to the same users. We analyzed the log of the processes and through the techniques studied, we analyzed where and when users click before you get to the purchase page. We then analyzed the processes that do not come to an end within a given period and / or number of steps. By reccomandation we are studying which supplier offers to show users depending on other offers seen and reported problems to their building administrator

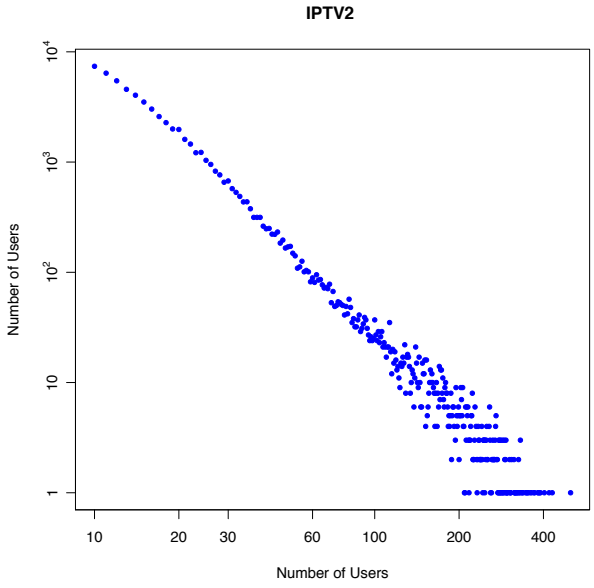
4.4.4 State of art

The major competitors of Condomani are desktop products (in Italy) and web application (outside Italy). Look at this desktop products: Danea Domostudio and Pigic MMdata are the most common management. These are on the market since 15+ years. These products are obsolete and are made on the model of Microsoft Access. All of these have bad usability. These products are for obsolete administrators, the same of which we complain. Anyway Danea was sold one year ago with a value of approximately 10 million Outside Italy are spreading products for the neighborhood and for the professionals. Among these, the social network of neighborhood NextDoor (raised 40M of funds), and the instrument of bidding MyHammer (stock market value of 15M), MyBuilder. Condomani is something more: a unique tool, managerial and social at the same time, which allows users to manage all the activities of our own building within a single tool.

Research side. None of the products on the market leverages innovative research topics that our product can benefit thanks to the studies discussed in this thesis.

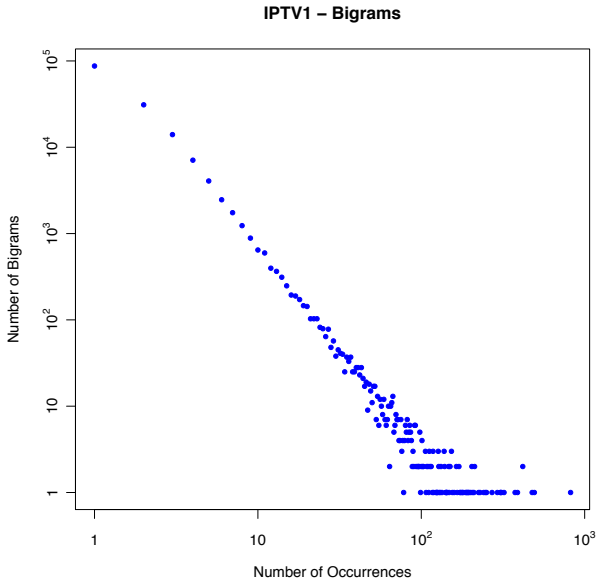


(a)

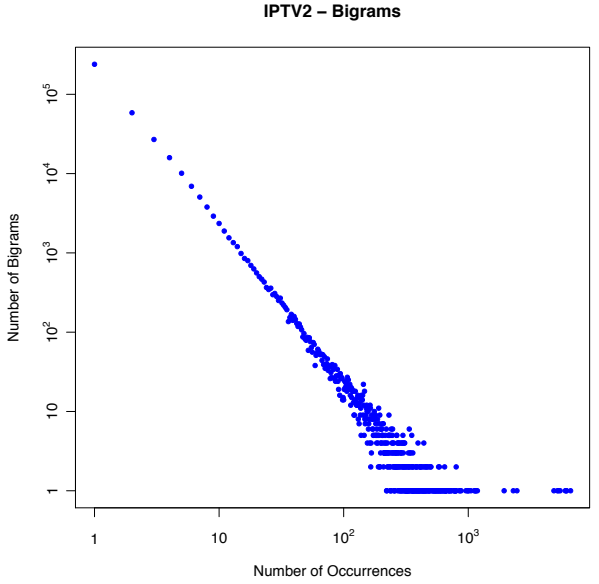


(b)

Fig. 4.2: Distribution of the number of evaluation per user on both datasets

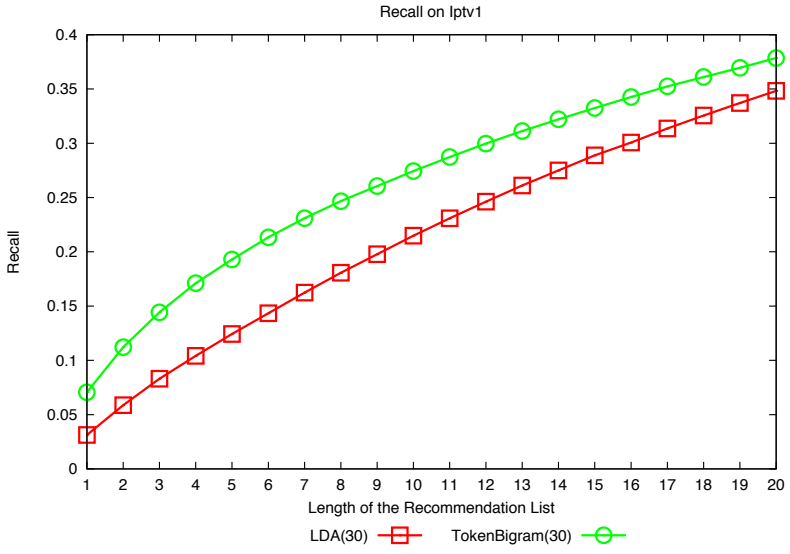


(a)

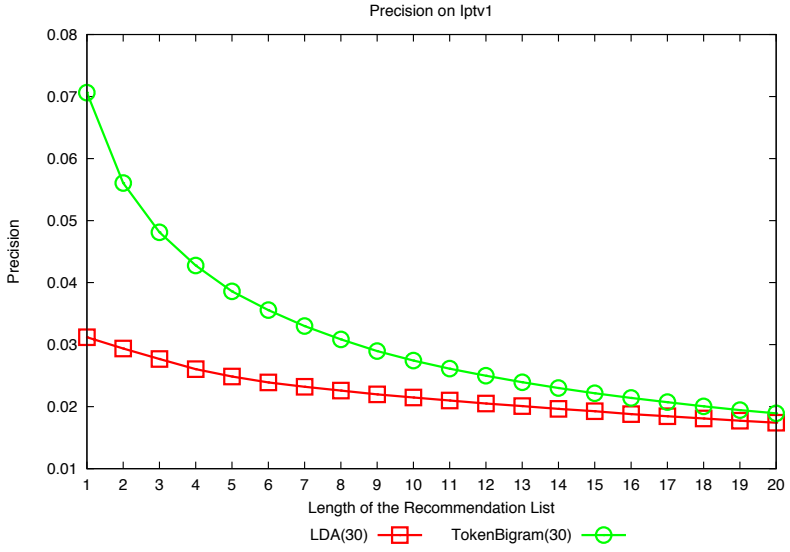


(b)

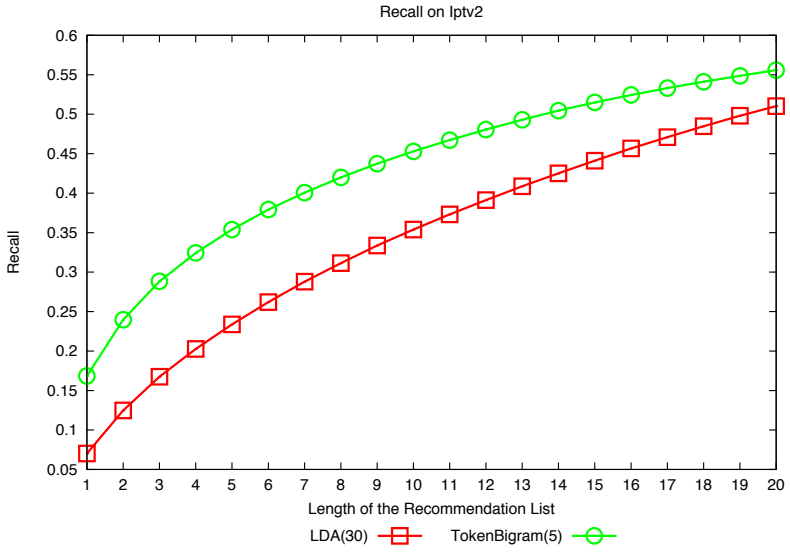
Fig. 4.3: Distribution of the number of bigrams on both datasets


Recall on Iptv1
Recall
Length of the Recommendation List
LDA(30)
TokenBigram(30)

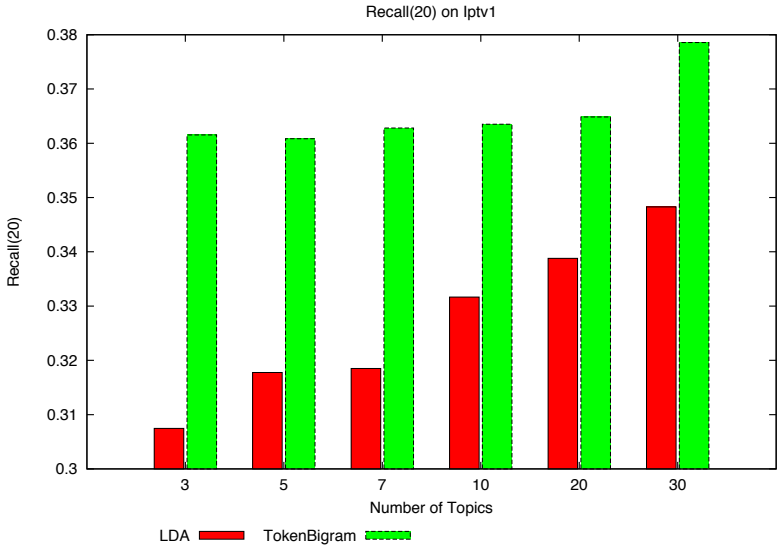
(a)


Precision on Iptv1
Precision
Length of the Recommendation List
LDA(30)
TokenBigram(30)

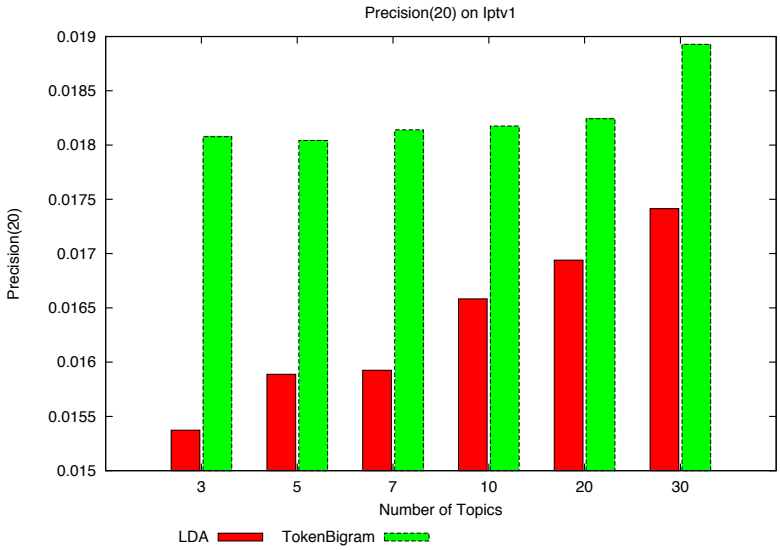
(b)


Recall on Iptv2
Recall
Length of the Recommendation List
LDA(30)
TokenBigram(5)

(c)

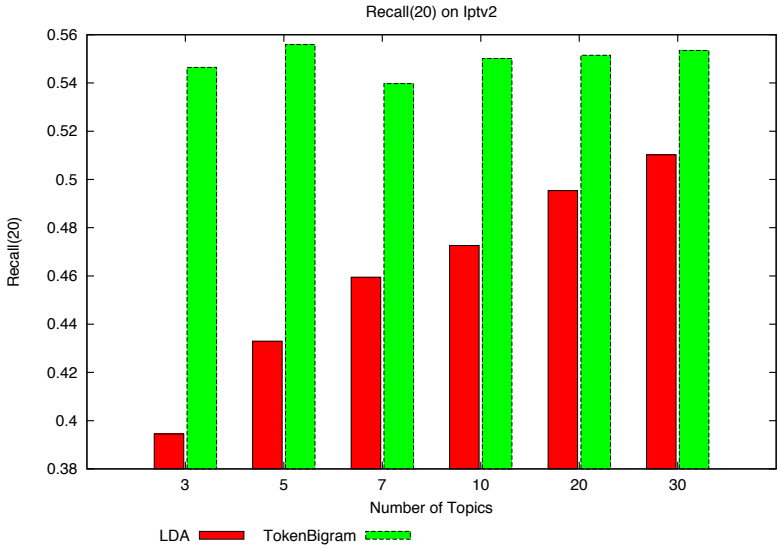


(a)

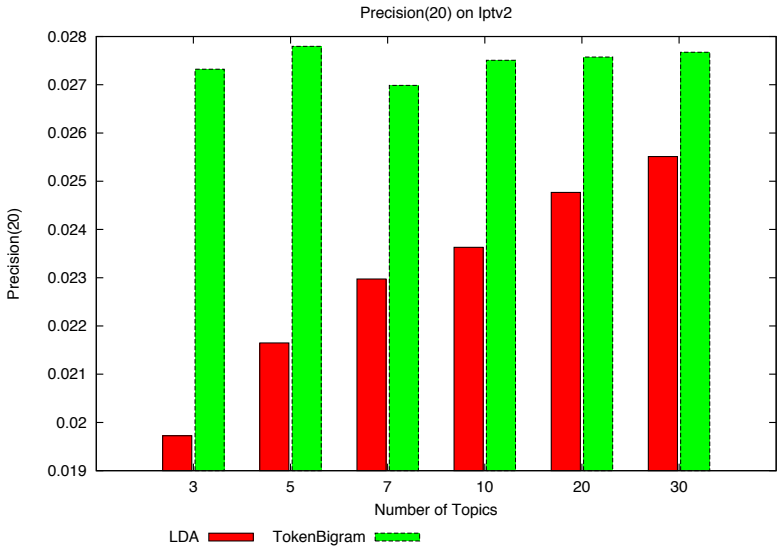


(b)

Fig. 4.5: Recall(20) of the considered approaches varying the number of topics on IPTV1

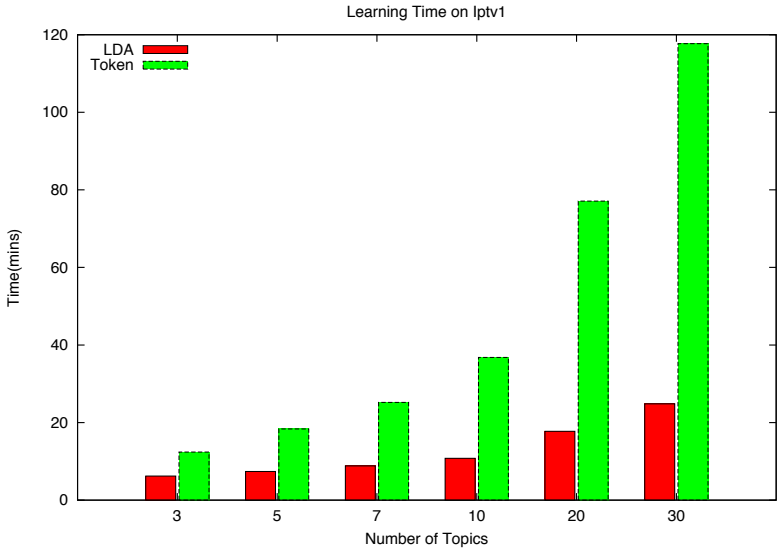


(a)

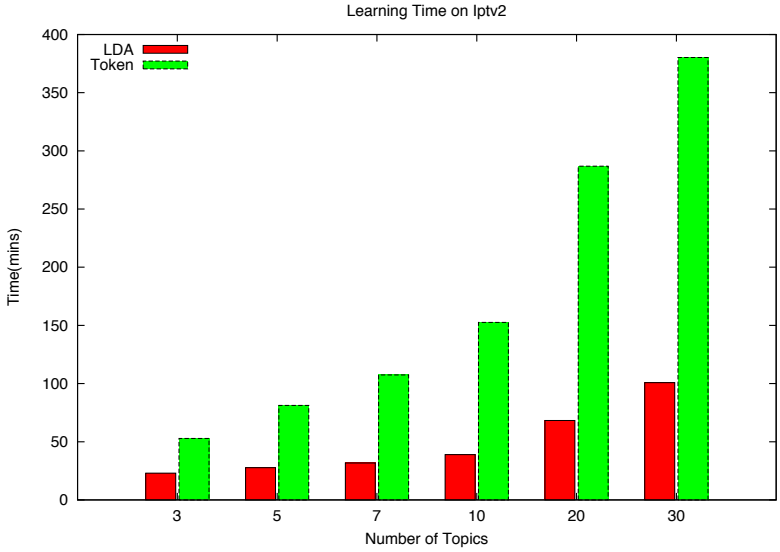


(b)

Fig. 4.6: Precision(20) of the considered approaches varying the number of topics on IPTV2



(a)



(b)

Fig. 4.7: Learning time of the models

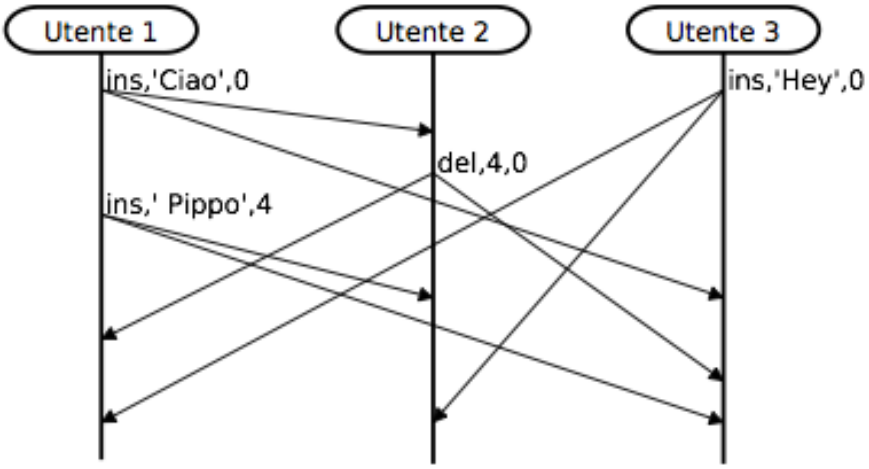


Fig. 4.8: example of Operational Transformation

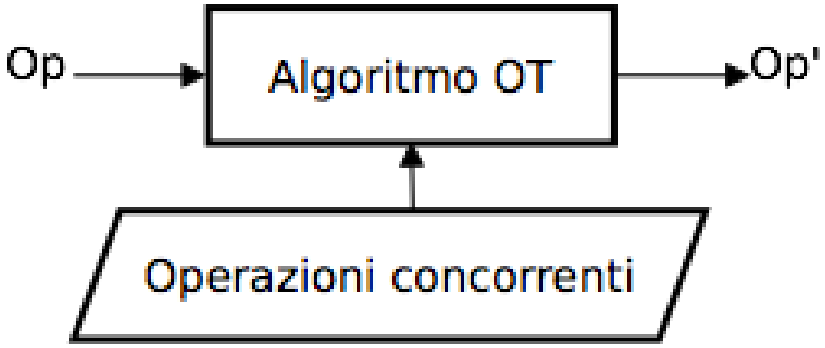


Fig. 4.9: System of Operational Transformation

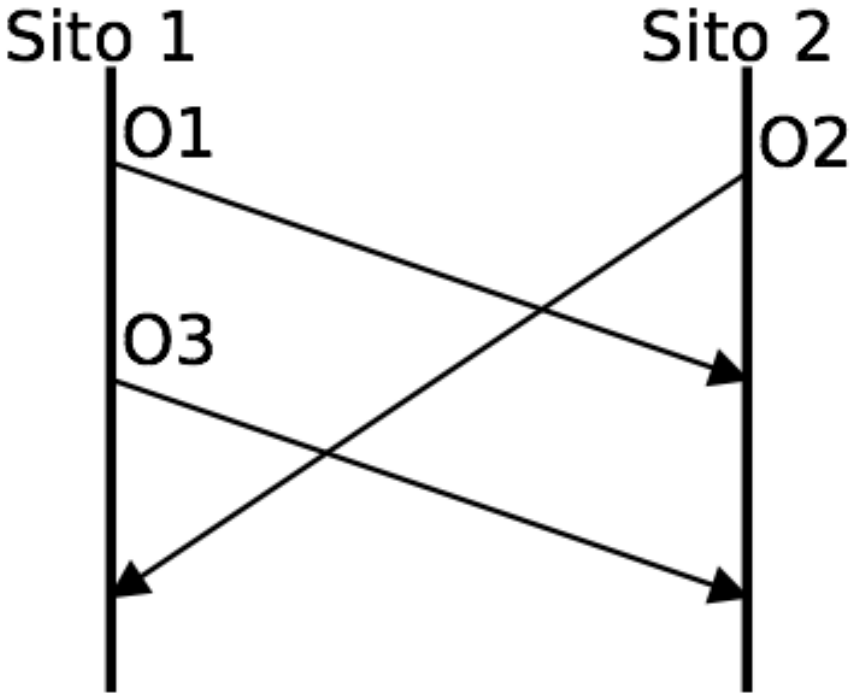
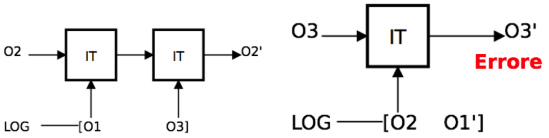


Fig. 4.10: The dOPT Puzzle



(a) algorithm dOPT, site 1 (b) algorithm dOPT, site 2

Fig. 4.11: Transformations on thesites of dOPT puzzle

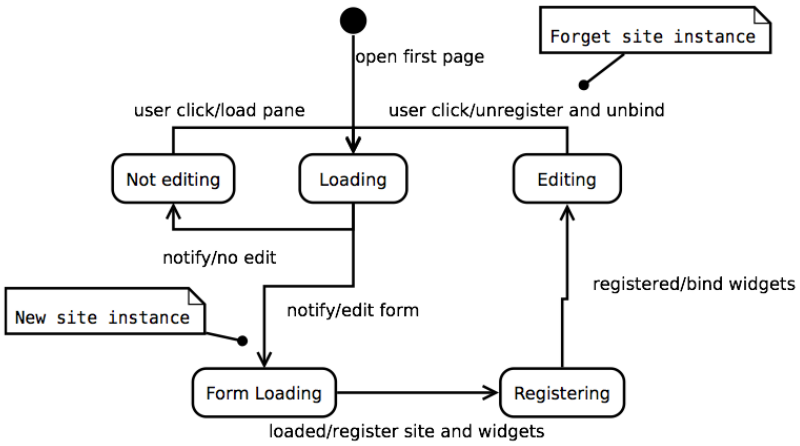


Fig. 4.12: life cycle and widgets

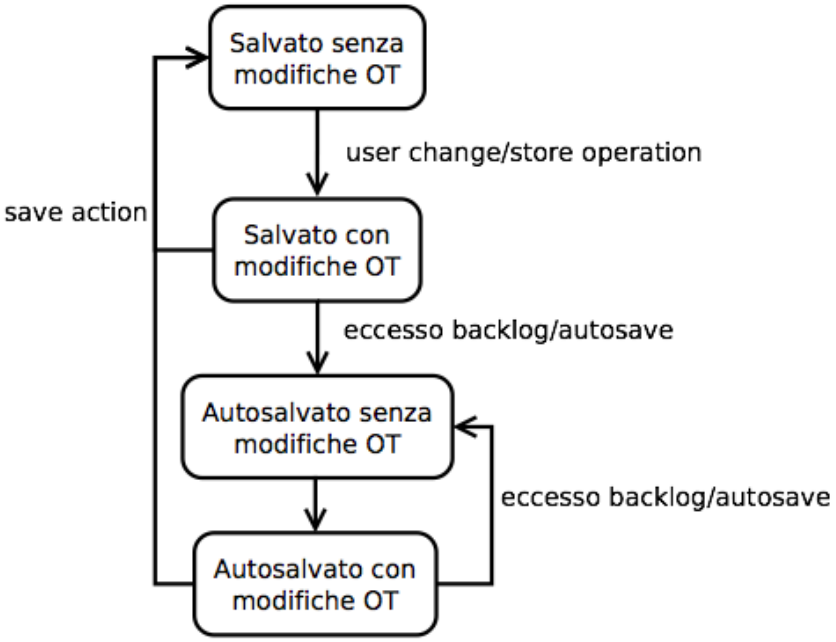


Fig. 4.13: States of an editing session

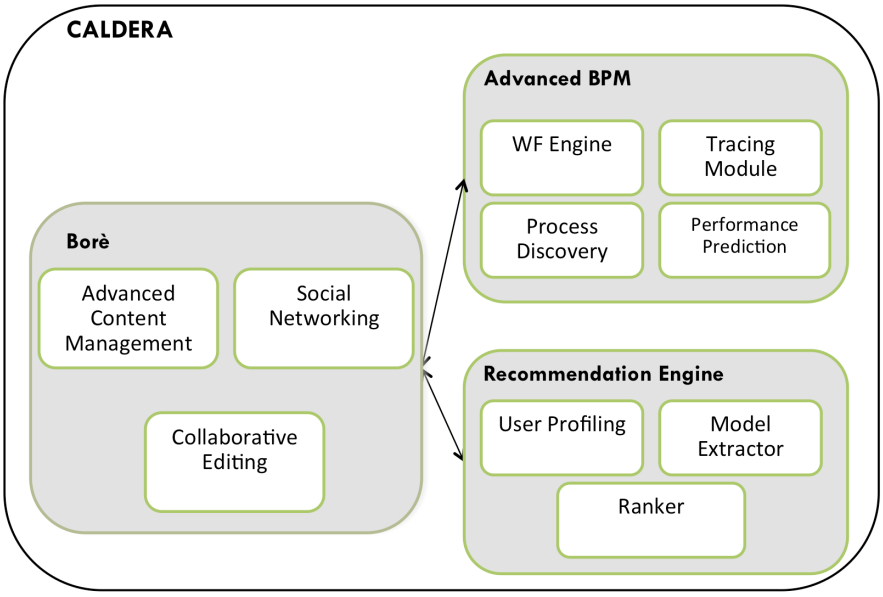


Fig. 4.14: Caldera platform

5

Conclusion

During my PhD studies we worked on a platform for the *web 3.0*. *Caldera* (Figure4.14).

In this work we proposed the development of an architectural paradigm for content-based web applications and focused on the cooperative interaction, whose foundations are formed on the principles of the Web3.0. Specifically, the goal of our work is to analyze issues related to the introduction of Web 3.0 in organizations and to address the problem of designing and implementing an architecture that enable organizations to adapt well to the characteristics of Web 3.0.

The main result is the creation of a platform which provides the user with the external environment in style of social networks where he can easily and flexibly manage all the activities of processes in which he is involved and also he will interact through the proposals that the system can provide. The user can benefit from the framework potential including access to it from a mobile device.

The main output of the project is a newly developed prototype platform called *Caldera*, which provides the Document Management System (DMS) for the management and analysis of collaborative processes and documents. The platform will be equipped with a powerful and varied set of tools, which offers a wide range of functionalities.

The framework is

- oriented to the model of social networks and based on cooperation between groups and individuals;
- integrated with intelligent functions for data analysis and suggestion;
- able to define and manage structured content even to the end user (including versioning, sharing, etc.);
- scalable, flexible, and extensible.

The strengths are given by these keywords:

- Document / Content Management;
- Social Cooperation;
- Workflow Management;

- Collaborative Editing;
- Recommendation;
- Knowledge Extraction;
- Process Mining.

Caldera is a platform for the collaborative management and analysis of processes and content.

Caldera is the evolution of the our platform *Borè*. *Borè* was already prepared for advanced content management and social networking. Caldera is an innovative platform for different reasons, including: (i) the ability to define and manage structured content; (ii) integration with the paradigm of social networking and cooperation; (iii) use of innovative models for data analysis and recommendation; (iv) analysis of execution traces and integration with innovative features of process mining.

The main contributions of this thesis are related to the following topics: Content Management, Social Cooperation and Collaborative System, Recommendation systems and Process Mining.

Document / Content Management and Social Cooperation.

We have proposed *Borè*, the development of the new architectural paradigm for content-based web applications founded on Web3.0 principles. The platform allows to reach high levels of customization; facilitates and enriches the web browsing experience of the users.

The novelty is summarized in the following list: (i) definition of new web resources; (ii) definition of mechanism of viewing, querying and storing resources; (iii) definition of events and actions associated to the resources; (iv) generation of social networks; (v) analysis of resource and relation data for processes of knowledge discovery.

Borè is extremely innovative in three dimensions. Foremost, it allows to define, organize, store, query and display the Web information as customizable objects and relations: an inexpert user can simply create the required Web. A second interesting feature of *Borè* is the possibility of directly supporting social networks (Social Cooperation), which spontaneously arise through user resource sharing. Finally, *Borè* allows the analysis of users' interaction with the published information by means of intelligent tools, that extract which recovered information on their interests, preferences and tastes from the observed interactions and use this exploit such information to customize and enrich their browsing experience.

Process Mining.

For Process Mining we deal with organization business processes. Our approach starts from the log analysis. Then we build decision trees related to different scenarios of execution. Any new process in running case is assigned to the correspondent cluster. The prediction can be performed using the model related on the selected cluster.

We have presented a new predictive process-mining approach, which fully exploits con- text information, and manages to find the right level of abstraction on log traces

in data-driven way. Combining several data mining and data transformation methods, the approach allows for recognizing different context-dependent process variants, while equipping each of them with a separate regression model.

The experimental results in a real application scenario are encouraging, showing that the method is precise and robust enough, and it does not require human intervention. Indeed, it is sufficient to use extreme values for the threshold support to have low prediction errors independently from the other finer-grain parameters (i.e., *maxGap* and *kTop*).

The technique has been integrated in a performance monitoring architecture, capable to provide managers and analysts with continuously updated performance statistics, as well as with the expected notification of possible SLA violations, which can be possibly prevented via suitable improvement policy.

This framework allow the organization using *Caldera* to analyze the process at runtime.

Recommendation systems.

For recommendation systems we are extending the popular Latent Dirichlet Allocation model by relaxing the bag-of-words assumption. The experimentation phase has proved its effectiveness. We have defined three new models. Shortly, in this work we have shown just one of them.

The proposed xtension of the LDA model relaxes the bag-of-words assumption of LDA, assuming that each token in not only depending on a number of latent factors but also on the previous token. The set of dependencies has been modelled as a stationary Markov chain, which led us to define a procedure for assessment the model parameters exploiting the Gibbs Sampling.

This model better suites a framework for modelling context in a recommendation setting than LDA, since it takes into account the information about the token sequence. The experimental evaluation over two real-world datasets expressing sequence information shows that the proposed model outperforms LDA at the expense of the higher execution time when the number of the latent topics is large, as the number of parameters for estimate is bigger than in LDA.

For users of Caldera the benefit of the recommendation systems is enormous. In fact, when surfing on the portions of the portal (subsets of graphs made by *Borè*) they are suggested to visit other sections (other subsets of graphs).

Collaborative Editing.

Initially, we have introduced the technique Operational Transformation and described how it can innovate the processes of collaboration between users.

We conducted a study of the various algorithms and have proven to be proficient in supporting the group editing systems replacing traditional techniques.

We have designed and implemented an extensible framework for client-server architectures, analyzing aspects of both client-side and server-side, and how the various components, during working synergistically, are separated and therefore replaceable (decoupling & code reuse). This framework is applicable for group editing in any context of Web 2.0 and Web 3.0, on any type of object that has interaction with the

end user. Particularly, it emerged that SOCT3 is a good solution for client-server architectures.

Finally, we have integrated in Caldera the support for collaborative real-time editing of complex documents by the users of the portal. In this phase the problem of saving centralized shared document has been solved as well as the integration with the navigation system Web 3.0 of Caldera.

A Use Case.

Finally, with a great satisfaction we applied what has been studied in real case, *Condomani.it*. A product that has become reality through this thesis.

Future work

For Mining Process as future work, we are planning to explore the usage of sequence-like patterns (e.g., k-order subsequence) in order to capture the structure of a process instance in more precise (but still quite abstract) manner, as well as to integrate our approach with a real BPM environment.

In the future for Recommendation systems we are going to investigate more types of Markov chains expressing the sequence of the tokens. Moreover, we have planned to improve the proposed model by considering supplementary information such as tags or comments over tokens.

References

1. Suh, P., Ellis, J., & Thiemecke, D. (2002). Content management systems. Peer Information.
2. van de Weerd, I., Brinkkemper, S., Souer, J., & Versendaal, J. (2006). A situational implementation method for webbased content management systemapplications: method engineering and validation in practice. Software Process: Improvement and Practice, 11(5), 521-538.
3. Ceri, S., Fraternali, P., & Bongio, A. (2000). Web Modeling Language (WebML): a modeling language for designing Web sites. Computer Networks, 33(1), 137-157.
4. Koch, N., & Wirsing, M. (2001, July). Software engineering for adaptive hypermedia applications. In 8th International Conference on User Modeling, Sonthofen, Germany.
5. Baresi, L., Garzotto, F., & Paolini, P. (2001, January). Extending UML for modeling web applications. In System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on (pp. 10-pp). IEEE.
6. Souer, J., Van De Weerd, I., Versendaal, J., & Brinkkemper, S. (2007). Situational requirements engineering for the development of content management system-based web applications. International Journal of Web Engineering and Technology, 3(4), 420-440.
7. Robertson, James. "So, what is a content management system." KM Column, June 3 (2003).
8. Vidgen, R., Goodwin, S., & Barnes, S. (2001, June). Web content management. In Proceedings of the 14th International Electronic Commerce Conference (pp. 465-480).
9. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. Scientific american, 284(5), 28-37.
10. Peters and Becker (2009) Peters, I. and Becker, P.: Folksonomies. Indexing and Retrieval in Web 2.0. Knowledge & Information: Studies in Information Science 2009, De Gruyter/Saur.
11. Pink (2005) Pink, Daniel H.: Folksonomy. New York Times, December 11, 2005.
12. Cesario et al. (2004) Cesario, E., Folino, F., and Ortale R.: Putting Enhanced Hypermedia Personalization into Practice via Web Mining. In Proc. of Int. Conf. on Database and Expert Systems Applications (DEXA), 2004, pp.947-956.
13. Lars-Gunnar Mattsson (2003) Reorganization of distribution in globalization of markets: the dynamic context of supply chain management. Supply Chain Management: An International Journal, Vol. 8 Iss: 5, pp.416 - 426
14. Marin, D. and Verdier, T. (2003) Globalization And The New Enterprise. Journal of the European Economic Association, 1: 337344.

15. Polak, Petr, Robertson, David C. and Lind, Magnus (2011) The New Role of the Corporate Treasurer: Emerging Trends in Response to the Financial Crisis (December 12, 2011). International Research Journal of Finance and Economics, No. 78, 2011.
16. Wonil Hwang and Gavriel Salvendy (2010) Number of people required for usability evaluation: the 102 rule. Commun. ACM 53, 5 (May 2010), 130-133.
17. Wikipedia (2012) http://en.wikipedia.org/wiki/Workflow
18. Van Der Aalst, W. M., Ter Hofstede, A. H., & Weske, M. (2003). Business process management: A survey (pp. 1-12). Springer Berlin Heidelberg.
19. Leon, A. (2008). Enterprise resource planning. Tata McGraw-Hill Education.
20. Sumner, M. (2007). Enterprise resource planning. Pearson Education.
21. Umble, E. J., Haft, R. R., & Umble, M. M. (2003). Enterprise resource planning: Implementation procedures and critical success factors. European journal of operational research, 146(2), 241-257.
22. Anton, J., & Petouhoff, N. (1996). Customer relationship management. Prentice Hall.
23. Buttle, F. (2012). Customer relationship management. Routledge.
24. Payne, A., & Frow, P. (2005). A strategic framework for customer relationship management. Journal of marketing, 167-176.
25. Paivarinta, T., & Munkvold, B. E. (2005, January). Enterprise content management: an integrated perspective on information management. In System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on (pp. 96-96). IEEE.
26. Tyrvinen, P., Pivrinta, T., Salminen, A., & Iivari, J. (2006). Characterizing the evolving research on enterprise content management. European Journal of Information Systems, 15(6), 627-634.
27. M. La Rosa, P. Soffer, eds (2012) BPM 2012 International Workshops, Tallin, Estonia, September 3, 2012, Revised Selected Papers. Lecture Notes in Business Information Processing, Volume 132. Springer, Tallin, Estonia.
28. N. Barbieri (2012) Probabilistic Approaches to Recommendations. Universit della Calabria
29. A. Bevacqua, L. Bruno, D. Sacc (2010) Realizzazione di un modello di consistenza per l'editing collaborativo di documenti nella piattaforma Web Bore. Universit della Calabria
30. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors (2011) Recommender Systems Handbook. Springer, 2011.
31. Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram (2007) Google news personalization: scalable online collaborative filtering. In Proceedings of the 16th international conference on World Wide Web, WWW 07, pages 271280, 2007.
32. J. Ben Schafer, Joseph A. Konstan, and John Riedl (2001) E-commerce recommendation applications. Data Min. Knowl. Discov., 5(1-2):115153, January 2001.
33. Aalst et al. (2012) Process Mining Manifesto. Business Process Management Workshops. 169-194.
34. Tancred Lindholm (2004) A three-way merge for XML documents. In Proceedings of the 2004 ACM symposium on Document engineering (DocEng '04). ACM, New York, NY, USA, 1-10.
35. S. Balasubramaniam and Benjamin C. Pierce. (2002) What is a le synchronizer? In Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 98), pages 98108.
36. Mercedes Martnez, Jean-Claude Derniame, and Pablo de la Fuente (2002) A method for the dynamic generation of virtual versions of evolving documents. In Proceedings of the 2002 ACM symposium on Applied computing (SAC '02). ACM, New York, NY, USA, 476-482.

37. Muriel Bowie, Oliver Schmid, Agnes Lisowska Masson, and Bat Hirsbrunner (2011) Web-based multipointer interaction on shared displays. In Proceedings of the ACM 2011 conference on Computer supported cooperative work (CSCW '11). ACM, New York, NY, USA, 609-612.
38. Mens (2002) Mens, T., "A state-of-the-art survey on software merging," Software Engineering, IEEE Transactions on , vol.28, no.5, pp.449,462, May 2002
39. H. Skaf-Molli, CL. Ignat, C. Rahhal, P. Molli (2007) H. Skaf-Molli, CL. Ignat, C. Rahhal, P. Molli. New work modes for collaborative writing.
40. Liu, X., El Saddik, A., & Georganas, N. D. (2003, May). An implementable architecture of an e-learning system. In Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on (Vol. 2, pp. 717-720). IEEE.
41. Abar, S., Abe, T., & Kinoshita, T. (2004, March). A next generation knowledge management system architecture. In Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on (Vol. 2, pp. 191-195). IEEE.
42. Ramparany, F., Poortinga, R., Stikic, M., Schmalenstroer, J., & Prante, T. (2007). An open context
43. Innocente, V., Silvestris, L., & Stickland, D. (2001). CMS Software Architecture: Software framework, xservices and persistency in high level trigger, reconstruction and analysis. Computer Physics
44. Anderson, C., & Wolff, M. (2010). The Web is dead. Long live the Internet. Wired Magazine, 17.
45. C. A. Ellis and S. J. Gibbs. Concurrency control in groupware systems. SIGMOD Rec., 18:399-407, June 1989.
46. Maher Suleiman, Michele Cart, and Jean Ferrie. Concurrent operations in a distributed and mobile collaborative environment. In Proceedings of the Fourteenth International Conference on Data Engineering, ICDE '98, pages 36-45, Washington, DC, USA, 1998. IEEE Computer Society.
47. David A. Nichols, Pavel Curtis, Michael Dixon, and John Lamping. High-latency, low-bandwidth windowing in the jupiter collaboration system. In Proceedings of the 8th annual ACM symposium on User interface and software technology, UIST '95, pages 111-120, New York, NY, USA, 1995. ACM.
48. Saul Greenberg and David Marwood. Real time groupware as a distributed system: concurrency control and its effect on the interface. In Proceedings of the 1994 ACM conference on Computer supported cooperative work, CSCW '94, pages 207-217, New York, NY, USA, 1994. ACM.
49. Chengzheng Sun, Xiaohua Jia, Yanchun Zhang, Yun Yang, and David Chen. Achieving con- vergence, causality preservation, and intention preservation in realtime cooperative editing systems. ACM Trans. Comput.-Hum. Interact., 5:63-108, March 1998.
50. Chengzheng Sun and David Chen. Consistency maintenance in real-time collaborative graphics editing systems. ACM Trans. Comput.-Hum. Interact., 9:1-41, March 2002.
51. Chengzheng Sun and Clarence Ellis. Operational transformation in real-time group editors: issues, algorithms, and achievements. In Proceedings of the 1998 ACM conference on Computer supported cooperative work, CSCW '98, pages 59-68, New York, NY, USA, 1998. ACM.
52. Chengzheng Sun, Yanchun Zhang, Xiahua Jia, and Yun Yang. A generic operation transformation scheme for consistency maintenance in real-time cooperative editing systems. In Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration challenge, GROUP '97, pages 425-434, New York, NY, USA, 1997. ACM.

53. David Sun and Chengzheng Sun. Context-based operational transformation in distributed collaborative editing systems. IEEE Trans. Parallel Distrib. Syst., 20:1454-1470, October 2009.
54. David Sun, Steven Xia, Chengzheng Sun, and David Chen. Operational transformation for collaborative word processing. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW '04, pages 437-446, New York, NY, USA, 2004. ACM.
55. Nicolas Vidot, Michelle Cart, Jean Ferrie, and Maher Suleiman. Copies convergence in a distributed real-time collaborative environment. In Proceedings of the 2000 ACM conference on Computer supported cooperative work, CSCW '00, pages 171-180, New York, NY, USA, 2000. ACM.
56. Oswald Campesato and Kevin Nilson. Web 2.0 Fundamentals for Developers: With AJAX, Development Tools, and Mobile Platforms. Jones and Bartlett Publishers, Inc., USA, 1st edition, 2010.
57. Staephane Martin and Denis Lugiez. Collaborative peer to peer edition: Avoiding conflicts is better than solving conflicts. In Hans Weghorn and Pedro T. Isaias, editors, Proceedings of the IADIS International Conference Applied Computing 2009, 19-21 November, Rome, Italy, 2 Volumes, pages 124-128. IADIS Press, 2009.
58. Du Li and Rui Li. Preserving operation effects relation in group editors. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW '04, pages 457-466, New York, NY, USA, 2004. ACM.
59. Du Li and Rui Li. An approach to ensuring consistency in peer-to-peer real-time group editors. Comput. Supported Coop. Work, 17:553-611, December 2008.
60. Du Li and Rui Li. An admissibility-based operational transformation framework for collaborative editing systems. Computer Supported Cooperative Work (CSCW), 19:1-43, 2010. 10.1007/s10606-009-9103-1.
61. Abdessamad Imine. Decentralized concurrency control for real-time collaborative editors. In Proceedings of the 8th international conference on New technologies in distributed systems, NOTERE '08, pages 41:1-41:9, New York, NY, USA, 2008. ACM.
62. Soren Lassen David Wang, Alex Mah. Google wave operational transformation, 2010. http://wave-protocol.googlecode.com/hg/whitepapers/operational-transform/operational-transform.html, Version 1.1 July 2010.
63. James E. Harmon. Dojo: Using the Dojo JavaScript Library to Build Ajax Applications. Addison-Wesley Professional, 1 edition, 2008.
64. Claudia-Lavinia Ignat and Moira C. Norrie. Customizable collaborative editor relying on treeopt algorithm. In Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work, pages 315-334, Norwell, MA, USA, 2003. Kluwer Academic Publishers.
65. Aguido Horatio Davis, Chengzheng Sun, and Junwei Lu. Generalizing operational transformation to the standard general markup language. In Proceedings of the 2002 ACM conference on Computer supported cooperative work, CSCW '02, pages 5867, New York, NY, USA, 2002. ACM.
66. Rui Li and Du Li. A new operational transformation framework for real-time group editors. IEEE Trans. Parallel Distrib. Syst., 18:307-319, March 2007.
67. Bondy, A. and Murty, U.S.R. (2008) Graph Theory. 3rd Corrected Printing, Springer, 2008.
68. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S. and Matera M. (2002) Designing Data-Intensive Web Applications. Morgan-Kaufmann, 2002.
69. Scott, M. L. (2006) Programming language pragmatics. Edition 2, Morgan Kaufmann, 2006, p. 470 vikas

70. Mitchell, J. (2002) 10 “Concepts in object-oriented languages”. Concepts in programming language. Cambridge, UK: Cambridge University Press. p. 287, 2002
71. Hamilton, J . (2009) Perspectives: One Size Does Not Fit All. Retrieved 13 November 2009.
72. Fay, C., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A. and Gruber, R. E. (2009) Bigtable: A Distributed Storage System for Structured Data. Google. Retrieved 13 November 2009.
73. Bell, R., Koren, Y. and Volinsky, C. (2007) Modeling relationships at multiple scales to improve accuracy of large recommender systems. In KDD ’07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 95 - 104, New York, NY, USA, 2007. ACM.
74. Marlin, B. (2003) Modeling user rating profiles for collaborative filtering. In NIPS*17, 2003.
75. Barbieri, N. and Manco, G. (2011). An analysis of probabilistic methods for top-n recommendation in collaborative filtering. In Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part I, ECML PKDD11, pages 172187.
76. Barbieri, N., Manco, G., Ortale, R., and Ritacco, E. (2011b). Balancing prediction and recommendation accuracy: Hierarchical latent factors for preference data. In Proc. SDM12.
77. Bishop, C. (2006). Pattern Recognition and Machine Learning. Springer.
78. Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. The Journal of Machine Learning Research, 3:9931022.
79. Cadez, I., Heckerman, D., Meek, C., Smyth, P., and White, S. (2000). Visualization of navigation patterns on a web site using model-based clustering. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD 00, pages 280284.
80. Clauset, A., Shalizi, C., and Newman, M. E. J. (2007). Power-law distributions in empirical data. SIAM Re- views.
81. Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In ACM RecSys, pages 3946.
82. Cremonesi, P. and Turrin, R. (2009). Analysis of cold-start recommendations in iptv systems. In Proceedings of the third ACM conference on Recommender systems, RecSys 09, pages 233236. ACM.
83. Griffiths, T. L., Steyvers, M., and Tenenbaum, J. B. (2007). Topics in semantic representation. Psychological Review 114.
84. Gruber, A., Weiss, Y., and Rosen-Zvi, M. (2007). Hidden topic markov models. Journal of Machine Learning Research, 2:163170.
85. Heinrich, G. (2008). Parameter Estimation for Text Analysis. Technical report, University of Leipzig.
86. Minka, T. P. (2000). Estimating a Dirichlet distribution. Technical report, Microsoft Research.
87. Wallach, H., Mimno, D., and McCallum, A. (2009). Rethinking lda: Why priors matter. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, Advances in Neural Information Processing Systems 22, pages 19731981.
88. Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In Proceedings of the 23rd international conference on Machine learning, ICML 06, pages 977 984.
89. X. Wang, A. M. and Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In Procs. ICDM07, pages 697 702.

90. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: a survey of issues and approaches. Data & Knowledge Engineering 47(2), 237–267 (2003)
91. van der Aalst, W.M.P., *et al.*: ProM 4.0: Comprehensive support for real process analysis. In: Proc. of 28th Int. Conf. on Applications and Theory of Petri Nets and Other Models of Concurrency (ICATPN'07). pp. 484–494 (2007)
92. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Information Systems 36(2), 450–475 (2011)
93. Blockeel, H., Raedt, L.D.: Top-down induction of first-order logical decision trees. Artificial Intelligence 101(1-2), 285–297 (1998)
94. Conforti, R., Fortino, G., Rosa, M.L., ter Hofstede, A.H.M.: History-aware, real-time risk detection in business processes. In: Proc. of 19th Int. Conf. on Cooperative Information Systems (CoopIS'11). pp. 100–118 (2011)
95. DLAI Group: CLUS: A predictive clustering system. Available at *http://dtai.cs.kuleuven.be/clus/* (1998)
96. van Dongen, B.F., Crooy, R.A., van der Aalst, W.M.P.: Cycle time prediction: When will this case finally be finished? In: Proc. of 16th Int. Conf. on Cooperative Information Systems (CoopIS'08). pp. 319–336 (2008)
97. Draper, N.R., Smith, H.: Applied Regression Analysis. Wiley Series in Probability and Statistics (1998)
98. Folino, F., Guarascio, M., Pontieri, L.: Discovering context-aware models for predicting business process performances. In: Proc. of 20th Int. Conf. on Cooperative Information Systems (CoopIS'12). pp. 287–304 (2012)
99. Frank, E., Hall, M.A., Holmes, G., Kirkby, R., Pfahringer, B.: Weka - a machine learning workbench for data mining. In: The Data Mining and Knowledge Discovery Handbook, pp. 1305–1314 (2005)
100. Hardle, W., Mammen, E.: Comparing nonparametric versus parametric regression fits. The Annals of Statistics 21(4), 1926–1947 (1993)
101. Harlde, W.: Applied NonParametric Regression. Cambridge University Press (1990)
102. Quinlan, R.J.: Learning with continuous classes. In: Proc. of 5th Australian Joint Conference on Artificial Intelligence (AI'92). pp. 343–348 (1992)
103. Schonenberg, H., Weber, B., Dongen, B., van der Aalst, W.P.M.: Supporting flexible processes through recommendations based on history. In: Proc. of the 6th International Conference on Business Process Management (BPM'08). pp. 51–66 (2008)
104. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, Second Edition. Morgan Kaufmann Publishers Inc. (2005)
105. Barbieri N. , Manco G. , Ritacco E. , Carnuccio M. , Bevacqua A.. Probabilistic topic models for sequence data. Machine Learning, 2013, Vol. 93, n. 1, pp. 5-29.