Joaquim Pedro

Nogueira da Costa de Castro Fonseca

Web Competitive Intelligence Methodology

Master's Degree Dissertation

Tutor: Professor Doutor António Grilo (FCT – UNL)

Jury:

President:	Prof. Virgínia Helena Arimateia de Campos Machado
Examiner:	Prof. Ricardo Jardim Gonçalves

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro de 2012

Joaquim Pedro Nogueira da Costa de Castro Fonseca

Web Competitive Intelligence Methodology

Master's Degree Dissertation

Tutor: Professor Doutor António Grilo (FCT – UNL)

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro de 2012

Web Competitive Intelligence Methodology

Science and Technologic Faculty

New University of Lisbon

Acknowledges

I would like to thank Miguel Soares of Vortal for the pertinent suggestions to develop the case studies.

Marco Delgado and Sudeep Ghimire helped me to address the technological dimensions of the research and I would like to express my gratitude to both.

To my Family and my Girlfriend who understood my absence during times of more work.

Finally, I'm much grateful to Professor Grilo for the full support and constant availability to discuss ideas, solutions and suggestions that lead me to develop this work.

Abstract

The present dissertation covers academic concerns in disruptive change that causes value displacements in today's competitive economic environment. To enhance survival capabilities organizations are increasing efforts in more untraditional business value assets such intellectual capital and competitive intelligence. Dynamic capabilities, a recent strategy theory states that companies have to develop adaptive capabilities to survive disruptive change and increase competitive advantage in incremental change phases.

Taking advantage of the large amount of information in the World Wide Web it is propose a methodology to develop applications to gather, filter and analyze web data and turn it into usable intelligence (WeCIM). In order to enhance information search and management quality it is proposed the use of ontologies that allow computers to "understand" particular knowledge domains.

Two case studies were conducted with satisfactory results. Two software prototypes were developed according to the proposed methodology. It is suggested that even a bigger step can be made. Not only the success of the methodology was proved but also common software architecture elements are present which suggests that a solid base can be design for different field applications based on web competitive intelligence tools.

Competitive Intelligence, World Wide Web, Semantic Web, Ontologies, WeCIM

Resumo

A presente dissertação abrange preocupações académicas sobre inovações disruptivas que causam mudanças de valor no ambiente económico actual cada vez mais competitivo. Para melhorar as capacidades de sobrevivência, as organizações estão aumentar os seus esforços em matérias de valor organizacional menos tradicionais como o capital intelectual e o uso de inteligência competitiva. A recente teoria *Dynamic Capabilities* propõe que as organizações desenvolvam capacidades de adaptação para sobreviverem a inovações disruptivas e aumentarem a sua vantagem competitiva em fases de mudança incremental.

Tirando proveito da vasta quantidade de informação presente na *Word Wide Web* propomos uma metodologia para o desenvolvimento de aplicações de recolha, filtragem e análise de dados para que sejam transformados em inteligência (WeCIM). Para aumentar a qualidade na pesquisa e gestão de informação, é proposto a utilização de ontologias que permitem que os computadores processem dados de acordo com domínios de conhecimento.

Foram desenvolvidos dois casos de estudo com resultados satisfatórios. Dois protótipos de *software* foram programados de acordo com a metodologia proposta. É sugerido que se possa abranger um nível de integração superior. Não só se obteve um bom resultado na aplicação da metodologia como os dois *softwares* demonstraram partilhar uma arquitectura similar sugerindo-se uma base sólida no design de diferentes aplicações baseados na recolha de inteligência a partir da web.

Inteligência Competitiva, World Wide Web, Internet Semântica, Ontologias, WeCIM

List of Contents

List of Figures

List of Tables

Abbreviations

API - Application Programming Interface

BI – Business Intelligence

BV – Base Value

CI – Competitive Intelligence

CPV - *Vocabulário-Comum-para-os-Contratos-Públicos*

CSV – Excel File Format

DOM – Document Object Model

ERP – Enterprise Resource Planning

ETL – Extraction, Transformation and Loading

HTML – Hyper Text Markup Language

IT – Information Technologies

KM – Knowledge Management

NLP – Natural Language Processor

OIL - Ontology Interchange Language

PC – Public Contract

PDF – Portable Document Format

PT – Public Tender

RDF – Resource Definitions Framework

R&D – Research and Development

SM – Skills Management

TXT – Text File Format

URL – Internet Shortcut (file name extension)

VB – Visual Basic

WeCIM – Web Competitive Intelligence Methodology

W3C – World Wide Web Consortium

XML – Extensible Markup Language

Chapter I - Introduction

1.1 Scope

Companies face an environment of increasing competitiveness. Many theories try to characterize organization strategy in their market and their ability to survive. It is accepted that changes in technology are a serious threat. The most acknowledged theory in strategizing was first proposed by Porter (1979). The author proposed five factors from which companies should shape their strategy that would yield the best competitive advantage.

Why companies fail to survive is also a matter of great interest in today's academic research. Authors have classified big technology changes and general innovations as a disruptive change, capable of changing market equilibrium, displacing product value and contributing to companies arise and fall.

These two subjects collide in a way that good strategizing should prevent failure in a presence of a disruptive change. What should be the ideal organization's capabilities that prevent failure and yield great competitiveness? Studying companies' survival capabilities some authors proposed different strategizing theories. One of the most recent and most complete theories considers the capability of a company to adapt to a new environment. This adaptation capability is surely necessary when disruptive change occurs. Since the pace of changing is increasing (Kessler and Chakraberti, 1996; Sood and Tellis, 2005) those capabilities have to be dynamic and therefore, Teece et al (1997) defended that one company's survival capability is related to their dynamic capabilities. The theory addresses why and how companies should adapt.

Aiming to study one of the many factors of why a company fails to adapt or even detect the occurrence of disruptive change that may determine that a strategic move is necessary, this dissertation focus on data analysis in the biggest source of information, the Internet. So it is propose that companies could generate better insights on strategic position, listening to information available on web. This information seeking should be part of an active Competitive Intelligence and Knowledge Management strategy. Companies could better detect strong and weak signals that may support decision making, allowing then to gain competitive advantage and survive in the competitive global economy.

1.2 Motivation

Business' academic research always encouraged academics to study and propose theories and models that would help managers to better understand competitive environment of today's economics. Recent examples of sudden rise and fall of big organizations still surprises researches and generates value to projects that thoroughly generates better understanding of market power.

Along with the competitive environment, the web is unexplored field of vast information. How could organization take advantage of "free" information relying on web sites? How could it be done using fully automatized software procedures?

On the previous two questions relies our motivation to create business applications that would crawl internet and gather pertinent information to be transform on competitive intelligence with real value. This application would support top-level, strategic decision making and help organization to gain competitive advantage and shield them against disruptive change.

Ultimately it is propose a methodology to develop web competitive intelligence applications that would help organization surviving disruptive change and gain competitive advantage during phases or incremental change.

1.3 Proposed model

The proposed model is a methodology that helps the development of software with competitive intelligence capabilities based on data collected from the web. The methodology follows a strict work flow to help design the competitive intelligence tool, from objectives idealization, ontology creation and programing concerns.

It is proposed that this methodology allows efficient creation of such tools that represent superior information analysis culminating in a competitive advantage for organizations. These tools can help understanding market behavior, and competition performance.

The methodology is created not only for a single project purpose but also to keep open possibility of a multiple tool framework. Each tool should have a specific objective but common information and results can be share between them to enrich their capabilities in data analysis and decision making models.

1.4 Objectives

The present dissertation has the objective to conceptualize a framework for the development of decision making aiding tools based on the formulated business ontology. This framework focuses on Competitive Intelligence and Knowledge Management subjects based on semantic search of information sources.

Due to the wide scope and range of issues regarding organization management there were chosen two subjects for prove of concept. The idea is to create a rule based ontology which can characterize the organization and manage information created by a set of tools. The tools are responsible for information extraction, gathering, filtering and analysis, presenting data in a structure way to aide decision making. Since some decision models are widely spread and used, the tool can also be responsible for some steps of the data processing according to a chosen model. The ontology serves as an information aggregator, data base structure and reasoning of the collected data.

Another capability of the framework is its repeatability. This means, the base framework can be set to any given organization and maintain their function regardless of industry. This doesn't avoid an initial phase of implementation but guarantees a degree of universality preventing major changes in the methodology process.

The challenges for this task can be enumerated as:

- To conceptualize a framework which can manage information gathered from the web and store on an ontology based data set.
- Integrate tools responsible for simple tasks regarding extraction, managing, analyzing and presentation of data.
- Guarantee interoperability capability to ensure that multiple tools can operate over the same library of ontologies.

The first matter is the central business model ontology. It has to be sufficient simple and yield a unified view of the subject so it can characterize any organization in any industry. On the other hand, it must characterize the target organization with enough deepness that the information management makes sense to the context. The populated ontology would then fully contextualize the environment. Other challenges regarding ontology management will arise in

what concerns ontology re-use, versioning, change and alignment, compatibility with data base structures, artifacts formats, APIs, etc.

The framework has to integrate different aspects of organization management using captured knowledge in the ontology. It can integrate models for knowledge management and competitive intelligence and other arising matters in today's management theories, according to the organization context. The framework will work as a library of models used in management practices with a high level of integration with the business model. Again, the framework has to be suited to any potential organization.

Last but not least, available tools can be integrated in the framework/ontology to allow extraction, filtering, processing and presentation of data. In addition to the organization data bases, information extraction from external sources has to be automatized. The extracted information has to be validated, filtered and stored in a normalized way to guarantee interoperability between all components of the framework. These tools could be previously arranged to match a determined objective such as monitoring competitors' product prices, gather scientific information, scan social media content, etc. to feed decision models. These tools would work directly on semantic ground given by the organization ontology. A modular architecture of the framework would allow tools to be independent from each other's which enables deletion, editing, and adding tools over the framework.

The value of the framework is to provide a superior information management tool with automated and semantic capabilities that enhances the organization capability of gathering and process contextualized information. These could play and determinant role providing enriched dynamic capabilities and survival abilities of the company in the present of disruptive change events. During incremental change the framework will enhance competitive capability maintaining automated set of information management processes. For the formulated objectives we can propose a research question and correspondent hypotheses.

Research Question

How can we create simple, systematic and automatic process to gather information from the Internet to aid strategic decision based on competitive intelligence?

Hypothesis

Web-based competitive intelligence information with automatic gathering, filtering, search and transformation can be effectively developed combining crawlers, wrappers and ontologies.

1.5 Methodology

The development of this dissertation started with a research on a range of business subjects regarding disruptive change. This allowed understanding why companies fail to survive and how they could gain survival capabilities.

The research led to subjects such as Competitive Intelligence and Knowledge Management, subjects with increasing significance in today's managerial world. CI and KM helps companies to understand how they can gain competitive advantage and understand their intellectual capital to explore dynamic capabilities. It was also explored how companies could feed CI and KM models with large amounts of data from the web. Due to the vast amount of information that can be legally access through the web, it was suggested that computer procedures could gather, filter and compile information, perform analysis to transform information into intelligence that would have competitive value.

Research continued to find information technology practices that could meet our objectives. The study brought subjects related to knowledge management using IT concepts, such as ontologies to store information with semantic meaning and crawlers and wrappers to automatically search the web.

After studying the viability of the project, some organizations were contacted in order to propose subjects to develop software prototypes. Software development was conducted to two cases that best suited our objectives and became the two case studies of this dissertation.

1.6 Dissertation structure

The study first begins with a literature review. It is divided over the Chapters II and III to clearly divide two areas of research. The first is business related and it describes why companies fail and how strategizing theories have evolved to identify company's strategic capabilities. It is also introduce some concepts of competitive intelligence because it is the inherit function of the methodology. In the second part of literature review, in Chapter III, it is briefly described programing concepts necessary to understand the software functionalities and already similar solutions in the web.

Chapter IV is the methodology presentation. It is described a six step process to develop the software to accomplish a pre-determined competitive intelligence objective. For each step a subchapter describes objectives and the main challenges encounter.

Next, Chapter IV outlines the two case studies developed using the methodology. It is described the environment in each market and the software objectives. Each application is then described according to each case study characteristics. A brief conclusion regarding objectives accomplishment, additional features, multiple configuration possibilities, and potential market value is drawn. Future work and general conclusions are present in Chapter VI.

Chapter II – Competitive intelligence

The objective and function of the developed framework is based on a complex context of business concepts and theories formulated in the past decades. It can be differentiated between matters that justify the need and use of the framework from concepts used by the tools.

Information and knowledge management gain decisive importance to characterize organization value in contrast with simple and exclusive resources based view. These justify the need for the framework which in the other hand includes transversal issues in business management. In a technical view is also necessary to introduce concepts regarding ontologies and information concepts. Since the scope of this work is business management, it is introduced the technical concepts only briefly in the correspondent chapter. Finally, there is the need for the actual picture regarding current offer in this category of information technology tools.

Therefore, the business related literature review is composed of:

- Disruptive change *vs.* companies.
- Dynamic capabilities for a successful strategy.
- Strategic foresight for planning the future.
- Competitive intelligence for competitive advantage.

The subjects listed above, form a rational sequence of ideas needed to justify and build the framework. An ontology-based diagram (Figure 2.1) illustrates the relation between research subjects.

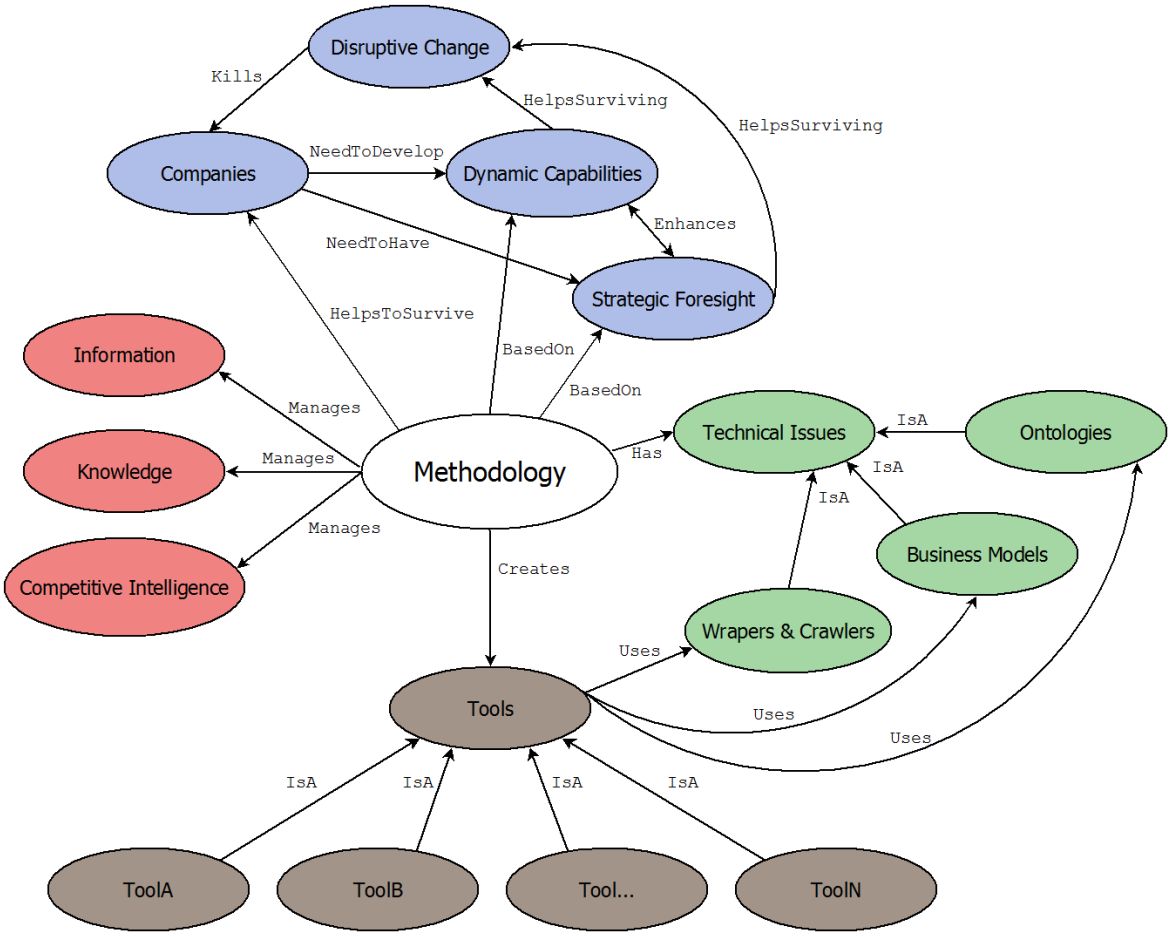


Figure 2.1 - Ontology based diagram illustrating subject's relations. Colors identified different subjects. Blue elements represent business concepts regarding disruptive change and strategizing. Red elements identify possible tools' functions. Green subjects are IT research areas and the tools are marked with the gray color.

2.1 Disruptive change *vs.* companies

Since it is proposed a methodology for organizations to enhance their competitive advantage and survival or adaptation ability, one must first understand the following question:

Why do large companies fail?

In 1980 digital cameras appear in the market and Kodak had already invented their product. A tremendous change had to be made on Kodak's business orientation. Kodak didn't pursue its invention because it was necessary to shift his high profitable film industry to a new digital one. Kodak had to

cannibalize their human resources and R&D structure from a chemistry based operation to electrical and physicist research areas (Deutsch, 2008).

Lucas and Goh (2009) classified the digital photography as a technological disruptive change that had a dramatic impact on film photography. The internal restructuring Kodak made as a response to the emergent competition in digital photography lead Kodak's workforce to fall from 145.000 to 27.000 employees between the 90's and 2007 while film photography market dimension was falling in the same rate digital was rising.

Many researchers studied Kodak case study and pointed reasons for their failure in digital photography. Lucas and Goh pointed the need for access threats and opportunities in information and communications technologies, Swasy (1997) considerer that Kodak's emphasis of doing everything according to the company's rulebook lead to adaptation and innovation failure, and Carly Fiorina interviewed by Batavick and Lucas (2008), mentioned that Kodak fail to understand the power of the consumers to shift from film to digital photography. No matter what was the key reason or bundle of factors for Kodak's downsizing and profit lost the cause is clear: a disruptive change. The new digital photography market claimed the substitution of film photography products.

Although changing factors can be very industry specific a broad definition of disruptive change can be exported from the Christensen (1997) view on *disruptive technology* as the appearance of products with a very different value proposition which often are *cheaper, simpler, smaller and more convenient to use*. The creation of these new markets or value networks disrupts current markets and value networks for a determinate period of time, displacing or erasing previous technology.

Christensen proposed five principals ideas for disruptive technology from which it was emphasized the following:

- Resourced dependent companies depend on customers and investors who supply those resources. Therefore customers and investors drive internal decision making.
- Large companies aim for large growth and small and emerging markets don't satisfy that need.
- Emerging and non-existing markets cannot be analyzed.

Rohrbeck (2010) on the other hand, summarized why companies had difficulties in their ability to adapt to changes in the environment. There were three main categories for failure:

- High rate of speed.

- Ignorance.
- Inertia.

High rate of speed

The idea that the rate of changes is increasing is well sustained by various researchers.

Kessler and Chakraberti (1996) argued that innovation speed is appropriate in environment of technology dynamism, intense competition and low regulatory restrictions. Organizations could be positively or negatively affected by innovation speed since it affected development costs, product quality and, therefore, project success. The idea of increasing innovation speed also sustains the shortening of product life cycles.

Sood and Tellis (2005) studied technology innovation and concluded not only that technology change is increasing but also the number of new technologies is increasing. They also discover that technologies will long stall periods without performance improvements have the biggest innovations steps when a change occurs. Another important idea is that new technologies can perform under or above current ones and their evolution is quite predictable but the secondary dimensions which create new and aggressive competition seemed random and unpredictable.

Another dimension of speed is given by the increasing diffusion capability of new technologies. Lee et al (2003) studied the impact of innovation on technology diffusion and conclude that radical innovation pressures diffusion speed. Therefor since technology innovation pace is increasing then diffusion speed is also increasing.

High rate of speed is also suitable for company's innovation pace. Different industries have different clock speeds and it affects the need for rapid innovation. Fine (1998) defended that organizations working on high clock speed environment were more exposed to rapid failure unless they developed effective adaptive capabilities and anticipate disruptive changes.

Ignorance

There are various reasons related to information management that sustains causes for companies' failure. They fail to sense and anticipate the occurrence of a disruptive change. The causes for this lack of information are document in the following researches.

Ansoff (1980) pursued the idea that technology innovation was faster than corporate strategic planning cycles which were coordinated with fiscal-year cycles. Organizations were failing in respond timely to disruptive changes.

Also the capture of announcing signals was failing. Various researches (Day and Shoemaker, 2006; Winter, 2004; Pinha e Cunha and Chia, 2007) claimed some signals are outside reach of corporate information scanning. Organizations focus on determined research areas and fail to capture information in peripheral sources.

Other factors such as the fact that information don't reach management levels capable of triggering a response or that information is filtered by middle management are also reported as responsible for ignorance about disruptive change signals.

Inertia

Rohrbeck (2010) pointed four reasons to justify the difficulty that large companies have to change their processes to adapt to a new environment. Companies have complex internal structures, are established in business networks with complex external structures, are not willing to cannibalized successful business lines, and don't pursue external technological breakthroughs. Facing a disruptive change, inertia disables companies' abilities to adapt. About disruptive change and companies' inability to adapt, Rohrbeck concluded that:

> "Large incumbent companies tend to be slow and ignorant and need to build dedicated structures for detecting and proactively managing discontinuous change".

We've seen that disruptive changes lead to business failure because they are not able to catch up with innovation speed, don't capture signals of an imminent change or are enable to adapt due to structure complexity. To change the company's ability to adapt, top level decisions have to be made. Therefore the company strategizing is at stake. The other element is time. Companies need to reorganize today to survive tomorrow. The combination of strategy with time is embraced in the *corporate foresight* concept.

But before, one has to understand the importance of intrinsic assets of a company and how it should shape companies strategy. In the past decades the idea that a company is much more than their capital and resource assets has gained importance. Managers start looking for knowledge and intellectual capital as an important parcel of the organization value. This may be due to one simple comparison between resources and knowledge – the second can't be bought.

Organizations need to focus in their expertise and develop difficult to imitate competencies. This idea leads us to *dynamic capabilities*, introduced in the next section.

2.2 Dynamic capabilities for a successful strategy

The most successful framework in the last decades to shape the organizations strategy and achieve competitive advantage was Porter's five competitive forces (Porter, 1979). This framework stated that companies should analyze five different forces that shaped the industry and defined how they could integrate and gain competitive advantage: i) *threat of new entrants*, ii) *bargaining power of customers*, iii) *threat of substitute products or services*, iv) *bargaining power of suppliers*, and v) *jockeying for position among current competitors*. Another theory was discussed by Shapiro (1989). He stated that game theory had emerged as the predominant methodology for managers to analyze business competitive factors depending on their competitors' moves and define the organizations strategy.

Teece et al (1997) defined the previous two theories as *models of strategy emphasizing the exploitation of market power.* They also defined resourced based theory as a *model of strategy emphasizing efficiency* that interprets firms with advanced structures and assets system which enable them to produce low cost or high quality products. Companies should manage scarce resources in order to appropriate economic rents.

They presented all the above theories to conclude that they all lack in competitive factors that shape successful organizations. Strategizing according to competitive forces and industry specific assets fail to pursue and create new sources of value. So they introduced the term *dynamic* as the ability to renew competences in order to adapt to changes in the environment. This ability should shape the firm's *capabilities* as the

> "…role of strategic management in appropriately adapting, integrating, and reconfiguring internal and external organizational skills, resources, and functional competences to match the requirements of a changing environment."

The proposition of Dynamic Capabilities is that successful companies *demonstrate timely responsiveness*, are *rapid and flexible in product innovation*, and gained management skills to *effectively coordinate and redeploy internal and external competences*. This proposed framework seeks to identify competences that are distinctive and difficult-to-replicate and, at the same time, represent the core of the company's competitive advantage.

In their proposal they explicitly differentiate what were market resources that can be bought and, therefore couldn't have any strategic value from capabilities that were firm specific and couldn't

be replicated. Teece et al (1997) defined three dimensions for those capabilities. They should built management *processes* constrained by their market *position* from a set of available *paths*. Each dimension was later described as a set of different definitions.

Other different definitions of dynamic capabilities can be grounded from literature. For example, Griffith and Harvey (2001) exposed them as:

> "Global dynamic capabilities are the creation of difficult-to-imitate combinations of resources, including effective coordination of inter-organizational relationships, on a global based that can provide a firm a competitive advantage."

Zollo and Winter (2002) express their view on dynamic capabilities as the organizational activity promoted to create new operational routines and adapted existing ones. Dynamic capability of a company develops with the co-evolution of three mechanisms:

- Tacit accumulation of past experience
- Knowledge articulation
- Knowledge codification processes

Cepeda and Vera (2007) summarized all the different definitions to extract four consensual ideas of dynamic capabilities:

1. The term capability refers to knowledge-supported organizational routines.
2. The starting point to develop dynamic capabilities is the current configuration of resources and routines.
3. Dynamic consists on evolution and transformation process of knowledge resources and routines.
4. The result is a different combination of the company's resources that yield a new configuration with superior competitiveness.

The main point in dynamic capabilities is that companies should develop a set of assets combined with knowledge and routines creation to produce a difficult to replicate or imitate value proposition. The way companies are starting to look to their assets in this ability to develop distinctive

competences is reflected in recent developments in other theories. How companies evaluate, manage and align intrinsic value is one of the main points for knowledge management and intellectual capital theories.

The linkage between dynamic capabilities and knowledge management is demonstrated by Eisenhardt and Martin (2000). Their view of dynamic capabilities reinforced the idea that the concept is based on *well-known processes, product development and strategic decision making*. In high rate speed environments dynamic capabilities consists in actual knowledge creation which results can be sustained by processes supported by routines based on previous knowledge. Managers' competences are enhanced when they reinforced existing knowledge but also when they promote new areas of interest.

This dynamic capabilities and knowledge management relation is even more evident in Cepeda and Vera view. They claim that with knowledge based transformation processes, managers are able to create, join and codify knowledge configuration which allow new improvements in the organizations routines. Hence, knowledge base improvements can trigger dynamic capabilities. The researchers also linked strategy dimensions to knowledge configurations stating that top managers can look to the organization's mission and values to influence knowledge configuration that most suits their strategy.

2.3 Information as a dynamic capability for competitive advantage

It was already discuss dynamic capabilities has a core competency that should drive management practices. But base on what? How can managers and decision makers base their policies? How can they even assess the need for change? What feeds their decision actions in order to achieve procedure changes?

Strategic foresight answers those questions in the management perspective: where to look, what to retain, how deeply, how often, but what is the pure raw material? In my view, information is the most important raw material to produce well founded and structured strategy. And information technology is the tool to work it. But this idea is not new.

Three decades ago Kantrow (2008) mention the need to incorporate the technology information discipline within the strategic decision making. Strategy and technology were unsociable. In 1983, Benjamin (1983) defined two basic IT drivers: the information technology economics - fast increasing processors and storage capabilities with decreasing production costs - and a new business environment

characterized by a global competition. The interaction between these two drivers caused what Benjamin called the *economic imperative of information technology*.

Just a couple years later, Porter and Millar (1985) explained how information could yield a competitive advantage to any organization with IT adoption and how game changing this technology would be. They mentioned changes in the company environment and processes from a supply chain perspective, product transformation and the direction and pace of change. Competition was also affected because IT would provide new business opportunities and all organizations could be influence by the following factors:

- Increased buyer power.
- Increased barriers for new markets duo to heavy IT investments.
- Computer aided design software would bring new substitute threats since products were becoming easier, cheaper and quicker to produce.
- Automated orders and procedures were causing increasing competition in distribution industries.

Information technology revolution took place years ago and nowadays every organization supports their processes based on IT infrastructures. Although IT infrastructures process information, most of that information is procedure related.

Today, all branches of an organization use and re-use information about cash flows, client related information, customer feedback, technological improvements, industry related scientific data, general communication inside house, etc. But it can be distinguish information usage in two different broad objectives: processes management and strategic foresight.

The first one is widely used and ranges from a variety of departments, objectives, tools, configurations and so on. The second is related to strategic decision and environmental scanning to ensure maximum capability to analyze and prepare for disruptive change, new business opportunities, adaptation. Information can be a characterized by the way company uses it. Rohrbeck (2010) described as the *elements* of information usage and how it plays important roles in his propose for a strategic foresight framework. The elements were already mentioned by other researchers and are as follows:

- *Reach:* Introduced by Reger (2001), this concept describes the way organizations scan external sources for business, related business and white spaces information.

- *Scope:* Mentioned by Becker (2002) and Jain (1984) as thematic areas and scope of scanning respectively, Rohrbeck divided them in four categories: *political, technological, consumer* and *competitive environment.* These segments can be related to the *reach* to access if the organization is scanning different environment in different depths.
- *Time Horizon:* as the named suggests, this element described by Becker categorizes the time frame in which information usage activities take place. The time horizon of information usage can be very briefly or to access strategic opportunities for the next decades.
- *Sources:* again, Becker and Jain mentioned the selection and usage of information sources. Reger refer that the most used sources were customers, suppliers and universities, each with different objectives.

We defend that IT infra structures have more potential. The information in organization is not being fully captured. The IT infrastructure can play a more profound role in information processes from internal and external sources. Today, IT is more a pathway to communicate that to process up to date, important and crucial data analysis tool which would support strategic changing decisions. My point is that one of the main goals of the framework is helping organization in knowledge creation and management. It supports the idea that those concepts assists one firms dynamic capabilities and therefor their survival, adaptation capability in a presence of a disruptive change and general competitive advantage.

2.4 Competitive intelligence for competitive advantage

It was already explained why companies need to develop procedures to collect and analyze information in order to became intelligence. This differentiation between information and intelligence is well discussed in literature but first, one needs to define what Competitive Intelligence (CI) is.

Miller (2001) exported the Society of Competitive Intelligence Professionals to define CI as the

> "the process of ethically colleting, analyzing and disseminating, accurate, relevant, specific, timely , foresighted and actionable intelligence regarding the implications of the business environment, competitors, and the organization itself ."

Authors had the necessity to clearly distinguish information from intelligence. Rouach and Santi (2001) reviewed that information is factual data and needs to undergo a process of filtering and treatment to become usable intelligence. Intelligence also differentiates from information by having a process value that managers need in order to undertake appropriate decision. It can be concluded that not only the form of the data – information is raw, intelligence is treated – but the value are distinguishable. Value can be extracted from information with the appropriate treatment from which it becomes intelligence.

The authors also classified a process to accomplish accurate CI. It starts by planning and defining a direction for a course of action. Collection of information and the correspond analysis to turn it to intelligence are intermediate actions. Dissemination is the end activity where intelligence should reach the appropriate management level.

The initial four step process is the base ground for two variation proposed by other authors. Fuld (2004) argue the presence of a storing activity and Ashton and Stacey (1995) add a sixth step regarding the whole process audit to classify the system's performance being predictable that it should sustain the system continuous improvement.

Competitive intelligence can yield competitive advantage in different ways. Rouach and Santi conclude in their research that those advantages cannot be denied. A good CI process implicates a timely detection of crucial information, help firms adopt good technology, and should characterize one firm's scientific and technical assets. According to the market, CI increases the chances of detecting threats and opportunities and clarifies the winning strategy in new market environments.

Chapter III – Web-based information management

Technical IT overview is necessary to understand some concepts addressed in this dissertation. In this section will present some notions of internet data, file formats and ontologies and a brief description of wrappers and crawlers.

Two examples of the technology applications are presented. One uses ontologies to manage knowledge in one organization and the second used internet extracted information in a competitive intelligence service. In the end of each case presentation it is discussed why the proposed methodology in this dissertation is a breakthrough against the existing examples.

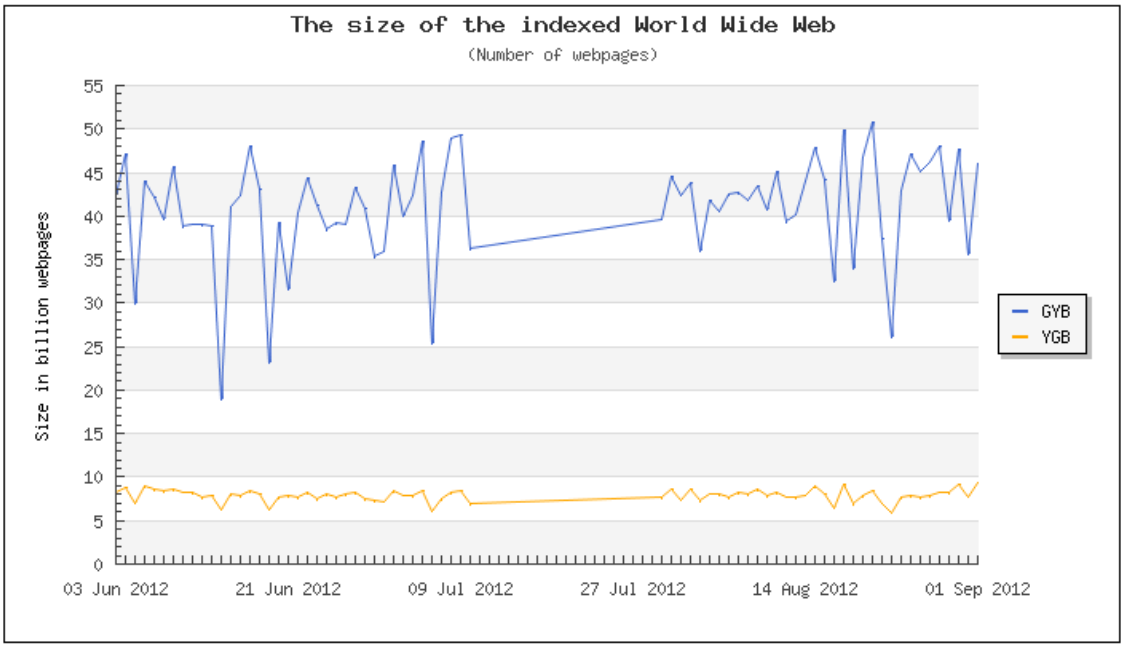
Therefore, technical review is divided in two sections:

- Soft technical issues:
  - The information overload and the semantic web solution.
  - Ontologies for semantic information management.
  - Information issues regarding interoperability.
  - Wrappers and crawlers for web interaction.
- The Competition: Web based tools for information and knowledge management.
  - Towards the semantic web - Ontology driven knowledge base: The Suice-Life (case study).
  - The Lixto simple approach.

3.1 Soft technical issues

3.1.1 The information overload and the semantic web solution

According to worldwidewebsize.com at the beginning of August 2012 the size of the internet is of 45 billion pages.



GYB = Sorted on Google, Yahoo! and Bing
YGB = Sorted on Yahoo!, Google and Bing

Figure 3.1 - Internet web pages according to worldwidewebsize.com consulted on 1-09-2012

How can an organization deal with such great amount of information and turn it into intelligence?

Internet being of huge size may not be an actual problem since target information can be restricted to few places but that may lead to a scope flaw and lead organizations to miss white spaces. Another question arises from information collection when it is simple humanly impossible to gather all data even when the target places are well identified. It is imperative that automated procedures can be able to handle and treat large amount of unstructured information.

Berners-Lee et al (2001) defined semantic web as:

> "The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation"

This new form of information categorization enables reasoning over web pages content and understands meaning without human intervention. Using metadata and top level ontologies software would perform far more complicated task using capture knowledge in the Internet.

So Semantic Web would be a response to nonstop growing internet. A serious of problems arises since Internet is a global culture and Semantic Web relies on the generalization of effort to categorize and correctly use metadata. Critics point out that people could misuse information categorization in order to mislead Semantic web engines.

Semantic Web is just starting and no information is intended to be collected based on semantic web features. But we sustain some of our work over semantic web protocols. It is proposed to use ontologies as knowledge representation of business domain; file formats used are the ones proposed by the W3C and provide an open possibility of interoperability and future integration with semantic web capabilities.

3.1.2 Ontologies for semantic information management

Strategic and competitive information analysis implies interpretation and contextualization of the organization. Unless information related process involves very strict technology based routines and data is consistently structured, those analysis and interpretations are done manually.

As we have previously discussed, general information lacks on semantic value. Technology fails to interpret value, meaning, and contextualization of data.

The oldest ontology definition derives from Philosophy and dates back two millenniums. Philosophers aimed two discuss the existence of things and how they could categorized them according to similarities in a hierarchy structure. This matter was intensively discussed and addressed, suffered from innumerous interpretations and applications that will not be discussed.

Recently, Guarino and Giaretta (1995) sensed the need of clarification. They came up with seven possible interpretations for the term "ontology" which they debated to find a clear definition. Regarding the philosophical definitions they defined it as *the branch of philosophy which deals with the nature and the organization of reality*. But in a practical formal way, ontology is:

> *"(...) (sense 1) a logical theory which gives an explicit, partial account of a conceptualization; (sense 2) synonym of conceptualization*."

In terms of knowledge engineering, a single ontology is just a simple conceptualization of one knowledge concept which forms a large group of ontologies that represent a larger knowledge base.

This definition is explicit in Guarino and Giaretta's view of *ontological engineering* as exploitation of *the principals of Ontology to build ontologies.*

IT ontologies can be classified according to two different factors. Gómez-Perez et al (2004) defined ontologies according to their internal structure and the actual subject of their conceptualization. They also classified ontologies according to their role and described four ontologies as being the most important ones:

- Knowledge Representation Ontologies.
- Top-level Ontologies.
- Linguistic Ontologies.
- Domain Ontologies.

The previous authors also described a group of methodologies to build ontologies. They can be more or less appropriate according to the ontology complexity, objective, and application environment. Some focuses only on the actual ontology definition while other extends the process to maintenance, re-use, and continuous improvement, or in other words, the ontology life-cycle. Next will be listed the methods described by the authors:

- Cyc
- Uschold and King
- Grűninger and Fox
- KACTUS
- MENTHONTOLOGY
- SENSUS
- On-to-Knowledge

The method that most suit the approach in this dissertation is building ontologies in Uschold and King's (1995) method. It is a very simple and straight forward and suits well our methodology since ontology building is just one of many steps to create one application. Uschold and King define a four step method illustrated in the figure 3.2.

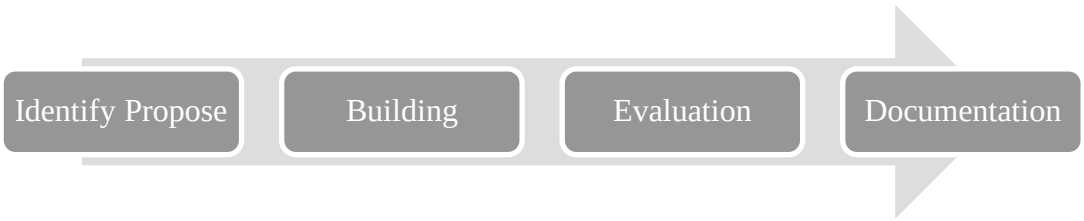


Figure 3.2 - Ontology building methodology (Source: Uschold and King's (1995))

Building phase includes capturing the right knowledge, coding the ontology and integration, if possible, with other ontologies. This simple methodology copes with our objective of software building. Documentation is also a good practice to track existing ontologies and evaluate integration possibilities with other software applications.

3.1.3 Information issues regarding interoperability

Interoperability is achieved in IT using diverse mechanisms in programing. Programing language is vast and varies greatly in performance and available capabilities. Usually a language is chosen if it best suit the intended objectives.

Another great development in the last decades of information history is the appearance of new file formats. These new file formats allow great levels of interoperability between different types of applications. According to Davies et al (2003) the XML creates a set of nested elements defined by open and closed tabs to create a tree structure of organized information. This markup language is user defined and allows solid data exchange between applications. Although XML would allow defining resources names a broader function is needed.

Using W3C recommendation of standard meta-data, XML gains also the capability of defining assertions meaning that relations between resources could be defined. This can be classified as Resource Description Framework (RDF).

With defined classes and assertions functionalities one more capability is needed to have a specially design format to create ontologies. This function is description logics - the ability to define restrictions to classes' proprieties. Ontology Interface Language (OIL) has this capability and thus turns a XML document, using predefine tags, into a tool which fully defines a knowledge domain, hence, to create a functional ontology file.

To conclude, XML markup language is a step further traditional file formats since it allows the user to define the tags meanings and create a meta-data language. Exporting standard tags and resource definitions is possible to add up capabilities to create a RDF file. Adding up description logics, this file type is capable of represent knowledge. Since application can reason according to those standard tags information is treated equally through completely different applications.

3.1.4 Wrappers and crawlers for web interaction

Web crawling is a program or script that scans web pages in a methodic and logic way. They can be used to index web pages, do market searches, find linguist terms, and are responsible for general web search.

Many examples of web crawlers can be listed:

- Yahoo! Slurp
- Bingbot
- Googlebot
- PolyBot
- RBSE

Contrary to web search, a wrapper is the program that deals with information mining. After crawling web, wrapper scripts are called to collect data. Some websites already implement a structured information capability that eases wrapper work. Unstructured formats are more difficult to handle since they lack an organizational scheme of information.

In information technologies these concepts often evolve to more specific definitions of web interaction. For instance web crawling is also defined as being a bot, web robot, spider and a web scutter. Wrapper challenges are in terms of handling information meaning, maintenance of their capabilities and manual labeling of collected information.

It is proposed to combine these concepts of web interaction using an ontology based store system that allows intelligence to be harvest from web data. Note that during the methodology and software development a more functional definition can be proposed to what crawler and wrapper functions really are.

3.2 The competition

3.2.1 The Swiss-Life case study

Reimer et al (2003) studied two cases of ontological application in real environment.

The first case study uses an ontology base application to undergo a Skill Management program. In their second case study, an ontology was built automatically from a set of 1000 pages document and was used to enhance search capabilities since some queries were not returning the intended results. Regarding the subject it will be discussed a few considerations over the first case study.

The ontology base application for skill management was created to undergo a knowledge management assessment of the firm. Ontology was formulated to be populated with the workers information. Information accounted general data, personal skills and job function, qualification, task and projects involved, and more personal information about interest and hobbies, among other data.

This management exercise would allow directors to search workers with specific skills, identify knowledge gaps, assess competency levels, structure training programs and document the firm intellectual capital assets. To develop this activity, a user interface was design to allow workers to introduce their information, and three ontologies were built to store data about skills, education and job functions, a specific query infrastructure was design and the whole system was evaluated.

During the development of the system the authors encountered some challenges that should be mention since they may contribute to a better understanding of ontology based applications. First they felt the lack of domain expert. The second difficulty concerns ontology evaluation. They could understand that some concepts and their relations where correctly formulated but couldn't evaluate the ontology integrity as a whole. For that they propose that users' feedback should contribute to evaluate the ontology when the application was already in use. The last problem is the size of the ontology that lead users to fail to find right concepts when filling the form or querying information. The solution is to use a dramatically smaller ontology. But a smaller ontology may lead to an insufficient use of concepts and, therefore, a loss of software capabilities.

From this case study, concerns arise for future work. Expert domains could be necessary if ontology concepts are fairly complex. Regarding small and very oriented software, this should not be a problem. Ontology evaluation should also be a proportional problem to the ontology size. Again, small oriented ontologies are more robust. Even so, application experimental phase should demonstrate how well the ontology is designed. The last problem is also an ontology size related problem. Complex solutions can be created using Natural Language Processing (NLP) and assessing the correspondence

between query language and the actual concepts names. During experimental phase, users behavior must be analyze to correct subjective nomenclature and ontology concepts may be redesign.

The Swiss Life case study demonstrates that ontology base applications are feasible to take part of the organization internal practices. There is a place within organizations to accept intelligence application that allow them to access their intellectual capital assets. It can be extrapolated that competitive intelligence applications could also be integrated in a company internal decision making process if they demonstrate enough value. Besides ontology base software, our methodology focus on information extracted from the web and is more than a simple analysis tool that uses an ontology store system to organize information. Since some competitive intelligence and knowledge management models sustain their analysis on web extracted information, our software goes a bit further than the Swiss Life Case Study.

3.2.2 The Lixto simple approach

Baumgartner et al (2005) reported on a simple tool for Business Intelligence consisting on web data extraction.

Lixto function serves as a Business Intelligence tool which they described as a process that provides closer *insight in a company and its chain of actions*. In this case, scanning competitors' information also copes with competitive intelligence definition.

The World Wide Web or simply Internet is full of information about competitors that can be legally accessed and register. However the observation and registration of that information is usually done manually. Technology allows computers to do individual task automatically but many problems arises: internet information is normally unstructured and is intended for human consumption.

As we have discussed, wrappers can make sense of the HTML structure of an Internet page to compile useful information. It has to be initially programmed to correctly identify how the web page is currently structure but, after initialization, the process is done automatically and information can be loaded to a XML document. Lixto goes a little beyond simple wrappers function with some additional features align to competitive intelligence. Lixto is used to compile product offers in a given web page, to monitor competitors' products' price or any other web based information.

First one must understand data paths in a BI system to apprehend the Lixto software working ground. Web data extraction is one of four data steps. Baumgartner et al divided data course as:

- **Data sources:** As the name suggests, is everything from where data can be extracted. This includes internal and external sources ranging from the company's DBMS or ERP to the World Wide Web and so on.
- **Data Integration:** How data can be transformed so their structure and formats are normalized. This is a key process since extracted data can range from a variety of different formats.
- **Data Storage:** How data can be organized and stored to be accessed, use, and re-used in the future.
- **Data usage:** Effective data usage - queries, analyzes, data mining, etc.

Those definitions are a bottom-up representation of the *Business intelligence reference process.* Lixto focus on the two bottom levels - data source and integration - by *extracting, transforming and loading* information. This three step tool is called an ETL tool. Figure 3.3 shows the business intelligence reference process with the identified Web-ETL site of action.

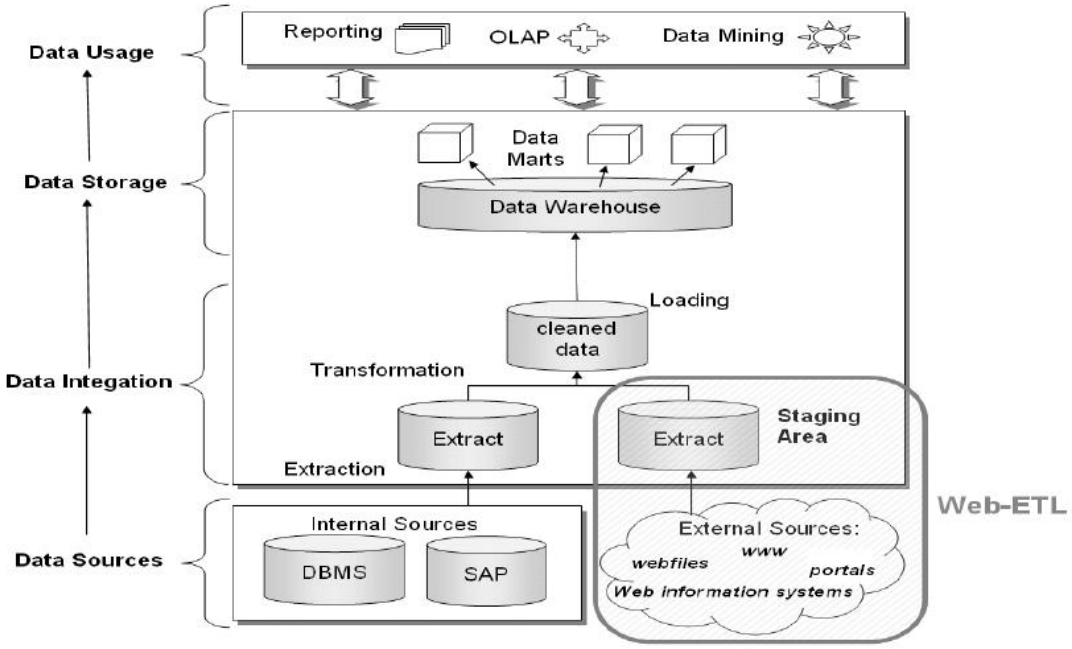


Figure 3.3 - The business Intelligence reference process (Source: Baumgartner et al, 2005)

Lixto software is made of different elements. An interface allows users to program different wrappers. The *Lixto Transformation Server* processes the wrappers automatically to retrieve

information based on events or a given schedule. The information is processed and loaded to the company's ERP such as a SAP service.

Lixto wrappers capabilities are advanced. They can record correct information even when the web page structures changes slightly; they can handle password protected log-ins; and gather information from overviewed pages.

The Lixto Transformation Server is also based on a user interface. This allows combination of different previous gather information of different wrappers on a single or multiple documents with structure information. The user can program different reports with cross wrappers' information scheduled for every hour or every morning. The information transfer protocols can interact with the organization's ERP as shown in figure 3.4.

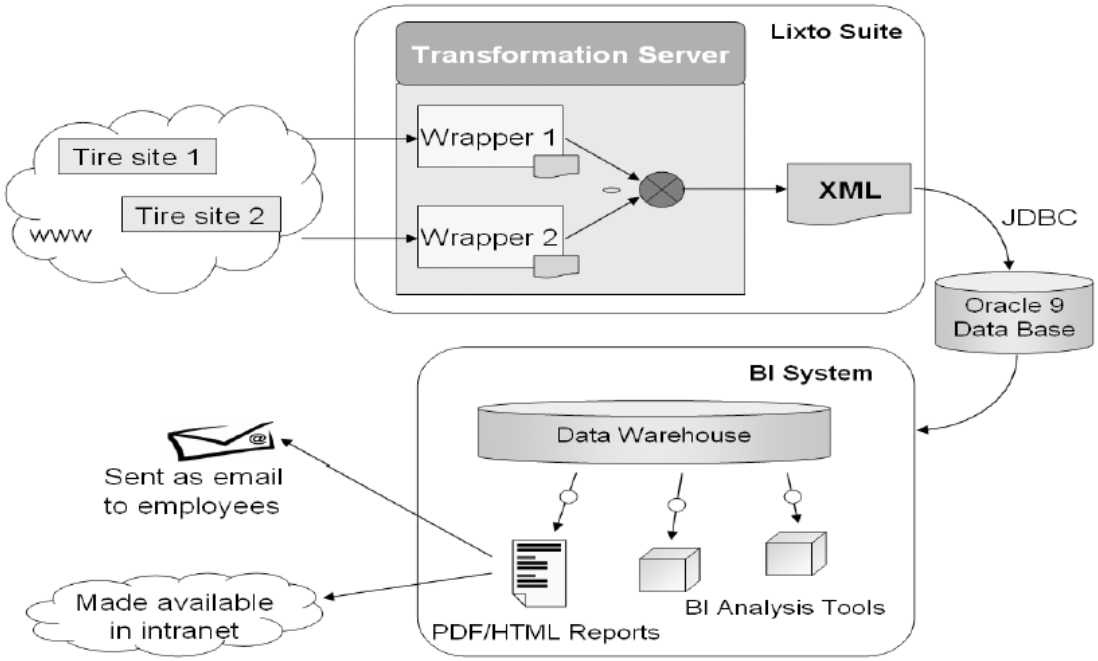


Figure 3.4 - Lixto architecture overview (Source: Baumgartner et al, 2005)

This software basically automatizes tasks. Searching and registering web data manually is not an added value activity and it is very time consuming. Besides automatizing extraction task, Lixto combines different extracted material in one or more reports. It has scheduling functions and can email information to managers. It can also be program to alert if a specified event occurs.

So, what is missing from Lixto? Why did we call it simple?

First Lixto is not working on an ontology based data warehouse which means there is missing potential. If extracted information can populate the ontology with semantic meaning, not only the information can be reported on time but also different tools can access the same information.

Lixto automatized most of the work but not all of it. If semantic ground is given, the ontology may already be populated with information. This means, the ontology “knows” competitors names, rival products, substitute products etc. The tool could analyze information structure and find where pertinent information is register.

Lixto stands alone. Lixto could function in parallel with other tools. In our context the extraction tool could stand alone for one type of reports but also it could be an extension and could be called by other software when they need to extract information.

Chapter IV – Web Competitive Intelligence Methodology

4.1 Introducing WeCIM

The basic objective is the creation of a simple methodology to develop, implement and manage Competitive Intelligence tools based on information collected from the web.

The tools consist in the interaction of four elements organized in a three component framework illustrated in figure 6. These concepts are correspondent with Lixto classification of data course: **sources**, **integration**, **storage** and **usage**. The crawler component deals with information **sources**; information is **integrated** by wrappers in an ontology based **store** system from where data analysis tools **use** the collected information.

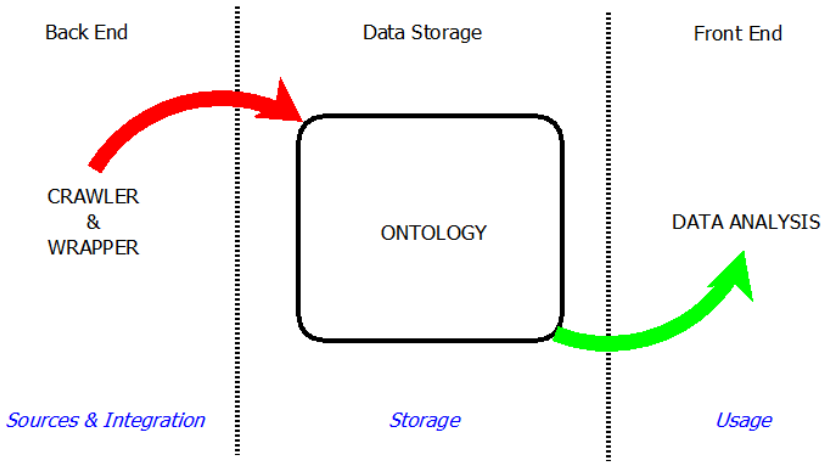


Figure 4.1 - WebCIM software elements

Back end refers to the software running while crawls websites and gathers information. Data storage is the data warehouse base on the ontology file. The front end is available to user interaction for data visualization. In the bottom is identified the data course presented.

Multiple tools can be arranged to manage a group of ontologies. Data analysis can also gather data from one or more ontologies. Hence, a multiple tool framework can be represented as:

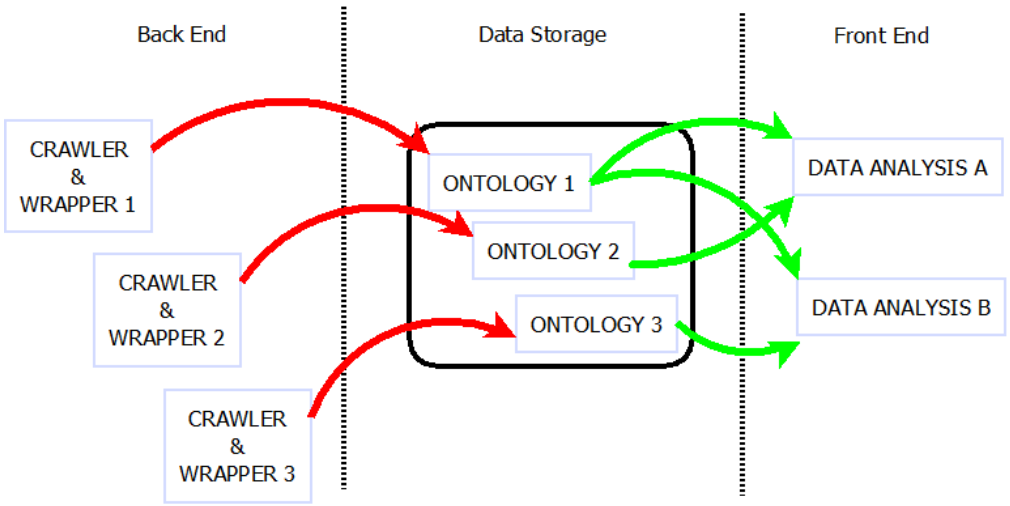


Figure 4.2 - Multiple software configuration

The proposed methodology is then responsible for the implementation of the multi component tool illustrated with a single or multiple web sources for data collection and analysis for competitive information management. The competitive advantage can be achieved by two means. Automatizing already establish data collection processes and to allow a broader, more complete, and humanly impossible data collection.

As described by the four information gaps that companies fail to pursue (Rohrbeck, 2010), all could be enhanced by the tool:

- Reach can be enhanced since automation allows more sources to be scanned. Even information sources that are not directly related to business competitive area can be scanned to enhance general information. White spaces can then be risk free monitored.
- Scope is enhanced with reach. Political, technological, consumer and competitive environment web sites can be monitored.
- Time horizon previous to the tool implementation depends on the web page side. Web page can keep old information and it can be accessible. During the software implementation old information can be fetched to complete a base data set. All information after tool implementation is saved on the ontology data base and the time lime is continuous, therefore, information is always available independently to the web site data set. This feature allows time trend to be analyzed.
- Sources are virtually infinite as long as they are present on the Internet.

4.2 Methodology development phases

Phases of development follow the four component of the framework.

Shortly, the tool methodology development goes over the next steps. Some can be worked in parallel but a simple precedency is presented that should guarantee the methodology integrity.

- Ontology building.
- Web page structure study.
- Crawling programming.
- Wrapper programing.
- Ontology population.
- Data analysis.

As it has been shown in technical issues literature review, ontologies building integrate requirements analysis. The requirements definition is directly related with the software objectives so it is considered that ontology building starts with the software objectives definition. These objectives, on the other hand, should emerge from the competitive advantage objectives. Note from the figure 4.1 that the ontology database store system is the central component of the framework. Hence, crawler and wrappers activities serve to populate the ontology and data analysis feed from it.

The next diagram illustrates tasks precedence's and relations. The effort of each task can vary in accordance to the structure complexity and software objectives. Automated procedures such as log in and search process in crawling activities increases software complexity.

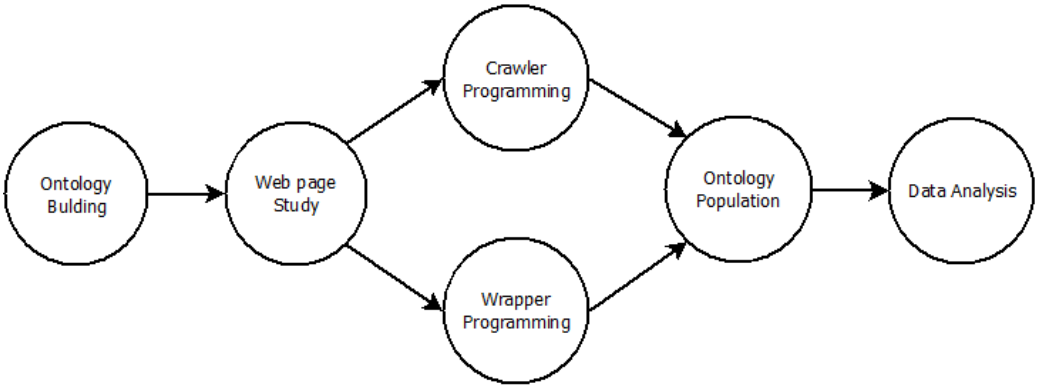


Figure 4.3 - Activities precedencies

Software and extensions used

For the target page HTML structure study it was used Google Chrome. Google Chrome features allow visualizing the real time HTML code of the page even when automated procedures change the page content. It is also crucial to unambiguously identify elements names.

For global programming it was used Visual Basic to program in Microsoft VB language. This is the base language to program windows applications. The compiled program uses also a set of given directories allocated in the C:/ path. There were two extensions used within the VB programming. The first to read and extract text strings from PDF files, called itextsharp.dll. The second was used to read and write XML/RDF ontology files according to its schema, called dotNetRDF.dll.

Another program was used to create ontologies schemas and control the integrity of ontologies output given by the program. To accomplish this task it was used Protégé.

After introducing task precedencies it can be correlated them with the programs used in each phase:

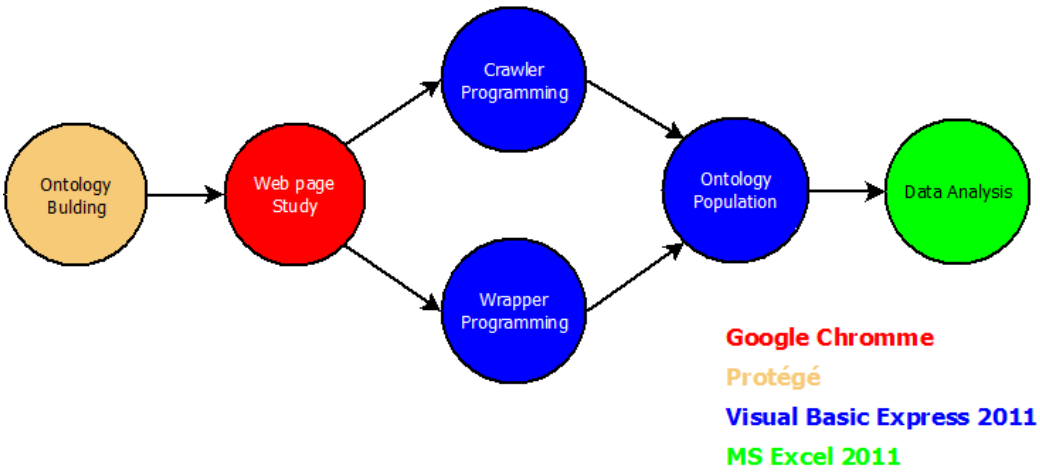


Figure 4.4 - Activities precedencies with proposed software

Data analysis was done in MS Excel in experimental phase for proof of concept but the author would recommend that data analysis would also be programmed in Visual basic creating a user interface where data visualization, search, query and general data analysis could be requested. Time constrains prevented the complete software development including the user interface for data visualization. However, a simple example of the ontology information visualization and simple filtering is shown in both case studies.

4.2.1 Ontology building

The ontology building methodology states that the process should start with the requirements definition. What would be the role of the ontology and knowledge field being represented? They inherit from the competitive objectives that the software proposes to accomplish.

Simple and focused objectives need simple ontologies schemas. Simple schemas are favorable to integrity control. Complex ontologies can be achieved by the re-use an integration of previously created ontologies. The multiple tool framework proposed with multiple software components working simultaneously on different ontologies can yield complex data analysis.

Hence, the first and most important step is the initial state of the software objectives. What is it proposed to achieve? What competitive advantage can be withdrawn from web available information?

Some typical objectives can be proposed provided that it is published on the web:

- Competitors products price monitoring.
- Raw materials price in stock markets.
- Public calls from international organizations.
- Transportations and other services strike warnings.
- Business related news.
- Exchange rates
- Competitors stock value
- Business related scientific papers.

All the above proposed functions for an application can somehow represent a competitive advantage and even support internal procedures efficiency and effectiveness.

As examples, knowing competitors prices trends allows determining marketing strategies and competitors business health. Raw material price trends can be understood and support buying decisions. Public calls from international organizations can advert for business opportunities and understand the evolution of market trends in public areas. Strikes information can help prepare production planning to mitigate their negative effect. Related news should advert for customers' perspective on brand names, international events and help marketing positioning. Exchange rates can be used to visualize trends and help financial management.

Most features are already available on internet services but the software integration would have much more potential since ontologies can be combined to correlate different kinds of information

using their semantic capabilities. Correlation between different types of data may allow a deeper understanding of market behavior and enhance competitive advantage of a company.

So, the first fundamental action of the proposed methodology is the definition of the key objective of the software. It is important to identify what information will be worked and how it can enhance competitive advantage within the organization processes and decision making.

With the requirements set the ontology definition should start. In this matter an iterative procedure is repeated until the final ontology is approved.

Classes, relations and standard individuals should be defined in order to correctly support two roles:

- Represent the knowledge field of the target information.
- Allow all data analysis proposed to gained competitive advantage.

Neither an incomplete knowledge should be achieved nor an incomprehensive one that is not capable of expressing meaningful conclusions.

Relations between classes should only be set if they express a meaningful function to the ontology. For instance, reasoning over the data only works if sufficient relations and restrictions are correctly set. If initially, no reasoning is predicted to be necessary, interclass relations and defined classes may not be used. This is important since, incorrect classification of classes may corrupt the ontology integrity.

The figure 4.5 represents the classes defined in the ontology using the Protégé software used in our first case study:

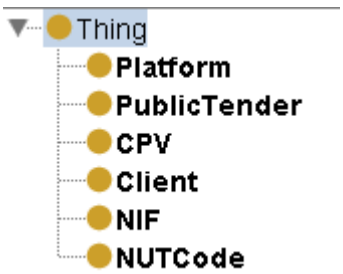


Figure 4.5 - Ontology taxonomy example

The presented taxonomy is simple and there are no parent classes. This resulted from the specific knowledge field in the case study.

Relations were used to organize classes' information structure. In figure 5.2 (Chapter 5.1.2) is illustrated the ontology representation of the previous classes with the respective relations and data properties. Classes are represented with the yellow filled circles. A line is drawn in-between two classes and represents the relations between then. All relations have to be named.

Some data about classes are not represented as classes since they are data representation of one class property. Classes are composed by single individuals that cannot be duplicated or else they will be inferred has being the same element. Data properties do not have individuals associated but only some kind of data: strings, integers, doubles or other. Two individuals can have the some value in some property without meaning they are actually the same individual.

From figure 5.2 it can be observed that the public tender, the main focus on the developed software, is the main element from where data and other classes are defined. Only strictly necessary classes should be considered. The CPV, Platform, Client, NUT's code and the data properties fully characterize the public tender in a way that analysis can be drawn with competitive value.

It is proposed that a time element should be always present. If the nature of the gather data has no time property related, it can be saved a data property containing the date when the information was gathered. The time property will allow a timeline visualization of different aspects of the collected data.

The establishing of a solid representation of the ontology sustains a good foundation for the future work. The classification of classes and standard individuals sustain the data needed to be gathered from the web to populate the ontology.

4.2.2 Web page structure study

The web page study serves two proposes: i) how the crawler algorithm is going to be program to access the target web page where information or documents are being hold; and ii) how the information in the HTML or documents is organized for the wrapping procedures.

Next it will be described key aspects in popular HTML schemas that help crawling and wrapping programming.

Crawling concerns

The first step is to understand where the target information is kept on the web page and how it has to be accessed. Information can be hold in various formats, can be structure or unstructured. Some sites already provide semi-structure formats like CSV or XL documents or even complete structure formats life XML.

The URL structure of a web page is very important to determine if the target page can be directly accessed or a serious of steps has to be performed. Another issue is the periodicity of the information. In some cases URL contains elements that indicate the date the information refers to, in other cases, the URL is static and only information is refreshed. It also has to be determined how refreshing procedures behave:

- Is information randomly changed?
- Does it have a correct and strict periodicity?
- Is old information kept available or is it lost?
- Target information is text to be interpreted? Or data is numeric and presented in tables?

These elements may determine crawling activities periodicity. Other questions must be understood in order to correctly program crawler features:

- Do search results appear in multiple URL or JavaScript's procedures are used?
- How the page behaves when there are no results to a keyword search?
- How do the crawler access if new information have been found?
- How does the server display an error page or a page with no information?
- Are any log-in procedures necessary?

Sometimes, in other to obtain the target page, log in or search procedures have to be done. It is very important to idealize a process that leads to the target location of the information accordingly to all questions arise before. Also, the algorithms must contain key checks if the correct path is being followed.

Next it will be presented a serious of examples showing methods to obtain the desire location of the information. Note that these studies determine the crawler behavior.

Simple URL construct over time variables

A simple URL construct is obtained using time variables and lead directly to the target web page. Knowing the URL structure is essential and changing key elements lead to the desire information. For example, in our first case study the target documents were held in a web page with the following structure:

```
http://www.example.com/newsoftheday/getnewsfromday=102.2011
```

It can be observed that changing the day index and year element, the web page with the corresponding documents of that date become available. In this first case the page's URL is well known and can be arranged to the desire time interval of the target information. Accessing the URL simply leads to the target web page where information or documents can be collected.

This kind of URLs can have different ways of representing the date or other identifiers. In this case procedures are similar but the input variables have to be programed accordingly. For instance, a similar URL with different elements has a similar solution:

```
http://www.example.com/newsoftheday/getnewsfromday=Jul-01-2011
```

In this case variables have a month, day and year parameters. The algorithm has to understand date structure and correctly construct the target URL. All month names have to be previously identified and testing may be needed to confirm the correct nomenclature.

No log-ins or search procedures are necessary. Only it is necessary to check if the page exists or if no information is presented. It is crucial, in study phase, to purposely provoke errors and understand how they could be detected by the crawler. Key text elements or changes in the URL must be monitored to detect an error presence from the web page side.

Simple server response to URL request

Another example of a straight forward access to information occurs when HTML is enriched with JavaScript procedures that request information to the web server. This was used in the second case study. In order to program the crawler one must understand how the HTML is behaving to keywords search.

A simple example is google.com web page. When a user searches for keywords the webpage itself constructs a URL. This URL requests the server the result page. Searching for the keywords "Portugal Lisboa" in google.com will result in the following URL:

```
https://www.google.pt/#hl=pt-PT&sclient=psy-
ab&q=Portugal+Lisboa&oq=Portugal+Lisboa(...)
```

There are a lot of secondary parameters being sent in the URL but the keywords are present. Manually changing those elements will result in a new search result. For instance, using the same URL but changing "Lisboa" to "Porto" will result in the search of the keywords "Portugal Porto" equivalent to the original Google search.

```
https://www.google.pt/#hl=pt-PT&sclient=psy-
ab&q=Portugal+Porto&oq=Portugal+Porto (...)
```

What this shows us a simple handling of the URL to do direct searches without using complex programing with web browsers.

Sometimes the URL is static or documents are automatically downloaded without letting the user see the actual URL that is being sent to the server. The possibility of directly download documents is welcome since these documents represent semi-structured information format that is easier to handle by the wrapper part of the software. Therefore it is the objective to request these documents in order to gather the target data.

When no URL is presented a more deep work is necessary. Using Google Chrome browser one can understand how the document is being requested. In the second case study this research was necessary. The following URL corresponds to the search page of base.gov.pt (figure 4.6).

```
http://www.base.gov.pt/base2/html/pesquisas/contratos.shtml?tipo=2#pes
quisa
```

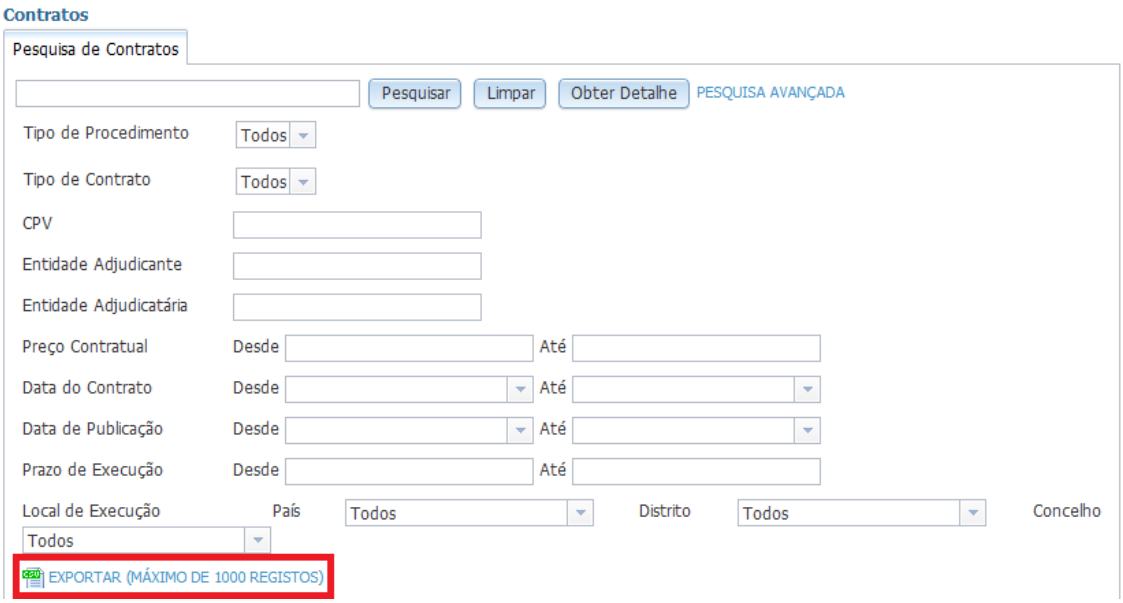


Figure 4.6 - Extract from the search page of base.gov.pt

Note there is the possibility of requesting an excel document (red rectangle, figure 12) with all the search results. There could be programed a complex procedure that would reach the web site, request the search page, fill the input boxes according to internal variables, and click the download button in order to download the document. But studying the JavaScript procedures of the web page, an equivalent procedure can be done by only constructing a simple URL.

For this, the element market by the red rectangle can be analyzed using a Google Chrome feature that presents the HTML code. Without inserting any search parameter, the excel document button is programmatically represented by:

```
<a href="/base/rest/contratos.csv?tipo=2#pesquisa"
dojoattachpoint="exportcsv" class="exportcsv > Exportar (máximo de 1000
registos) </a>
```

Note that the hyperlink has no search information whatsoever. If we are interested in all contracts from date 05/04/2012, it would be necessary to insert the data on the corresponding input fields and click the "Pesquisar" button marked with a red rectangle in the Figure 4.7.

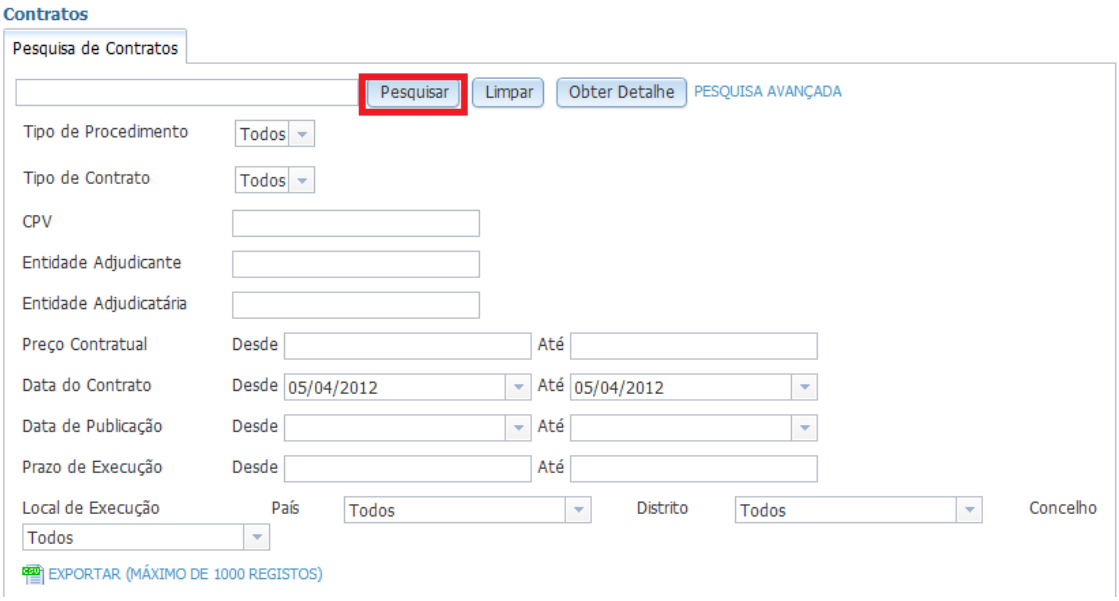


Figure 4.7 - Extract from the search page of base.gov.pt

After the results are presented in the page, one can inspect the same excel button element. Now the HTML code has changed to:

```
<a
href="/base2/rest/contratos.csv?texto=&amp;tipo=0&amp;tipocontrato=0&amp;cpv=&amp;adjudicante=&amp;adjudicataria=&amp;desdeprecocontrato=&amp;atepreco
contrato=&amp;desdedatacontrato=2012-04-05&amp;atedatacontrato=2012-04-
05&amp;desdedatapublicacao=&amp;atedatapublicacao=&amp;desdeprazoexecucao=&
amp;ateprazoexecucao=&amp;pais=0&amp;distrito=0&amp;concelho=0"
dojoattachpoint="exportcsv" class="exportcsv"> Exportar (máximo de 1000
registos) </a>
```

We can clearly identify a hyperlink requesting the document with our search specification in it:

```
="/base2/rest/contratos.csv?texto=&amp;tipo=0&amp;tipocontrato=0&amp;cpv=&amp;adjudicante=&amp;adjudicataria=&amp;desdeprecocontrato=&amp;atepreco
contrato=&amp;desdedatacontrato=2012-04-05&amp;atedatacontrato=2012-04-
05&amp;desdedatapublicacao=&amp;atedatapublicacao=&amp;desdeprazoexecucao=&
amp;ateprazoexecucao=&amp;pais=0&amp;distrito=0&amp;concelho=0
```

All search elements are present in this URL. The elements that were not filled have a corresponding variable with null value, and the date fields have values corresponding to the chosen input.

Adding the root element, manipulating the search parameters present in the URL, and finally checking the encoding, it is possible to directly query the server without doing any search procedures, simplifying crawling capabilities.

After this study the same simple URL procedure can be used. Since the page responds with a document a Client Web Document Download method can be used to download the document directly to a desired local directory.

Search, log-in and other procedures

Sometimes, the target page may request a user to be logged-in to access information. In other cases, the software will search for keywords or other procedures have to be done in the web page. Without the possibility of directly using URL handling these procedures have to be done programmatically. Both log-in and search procedures have two key elements associated:

- Filling one or more text boxes.
- Clicking a button to submit the form.

Technical review will be done in the Wrapper Programming Chapter. For the study phase is only necessary to gather the following information that can be collected using Google Chrome's features such as inspecting the involved elements.

Text boxes' names have to be identified to be used in the program algorithm. Also the texts that will be inserted have to be known and it has to be ensured that the user name and password work. For buttons to be clicked, after filling the input or search boxes, their names have to be also collected.

Wrapping concerns

A simple observation of the HTML code may be sufficient to fetch the desire information. Some factors may lead to simpler or more complex wrapping capabilities. These questions help understanding how wrapping functions should be programed.

- Does the amount of data vary?

- Is it looking for a single or a set of values?
- Is the target information always in a fixed place?
- What HTML elements commonly identify target data?
- Are there any conditions that determine whether the information is important or can be rejected?
- The crawling output is an HTML page or a document?
- Do the documents formats have to be changed before collecting the information?

All these elements have to be understood before programing any wrapper that will have a satisfactory behavior. Also remember that the wrapper prepares the information to be store in the ontology data base. That's why the ontology schema is the first step in the methodology.

We can also discuss the two-step procedure when reading text from a HTML or another text file. Most times, the software will read the target files line by line. When doing this action, the software first has to determine if the line contains desire information. If positive, it has to clean the information to the pretended format. For example, if looking for a price tag of the product X that is present in the next text line, the software would first check the existence of the word "Product X" and the words that indicate the price will be presented, for example "price" and "€":

```
<p> The Product X has a selling price of 1020, 00€ <p>
```

After positively checking the presence of the text elements, the wrapper has to clean the string to obtain strictly the price:

```
1020, 00
```

In this study phase it's necessary to identify all key elements that guarantee the correct collection of data and the text structure in order to clean unnecessary substrings.

Wrapping from HTML code page

If target information is present in a web page the crawler features should output one or a set of HTML documents. These documents can be downloaded into a TXT format. A simple representation of a HTML web page in text format is presented in the next figure.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="pt" xml:lang="pt">
<head>
        <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
        <meta name="author" content="Imprensa Nacional-Casa da Moeda, S.A." />
        <meta name="description" content="S&iacute;tio do Di&acute;rio da Rep&uacute;blica Eletr&oacute;nico" />
        <meta name="keywords" content="DRE Di&aacute;rio Rep♦blica Eletr♦nico Legisla&ccedil;&atilde;o INCM S&eacute;rie I :
        <meta http-equiv="cache-control" content="no-cache" />
        <meta http-equiv="pragma" content="no-cache" />
        <meta name="robots" content="index,follow" />

        <title>Parte L - ♦ltimos 45 Dias da 2.♦ S♦rie (Servi♦o por Assinatura) - D.R.E.</title>
        <link rel="stylesheet" type="text/css" href="/comum/css/geral.css" media="all" />
        <link rel="stylesheet" type="text/css" href="/comum/css/sug.css" media="all" />
        <link rel="stylesheet" type="text/css" href="/comum/css/listas.css" media="all" />
        <link rel="stylesheet" type="text/css" href="/comum/css/print.css" media="print" />
        <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
        <link rel="prev" href="/sug/2s/udr.asp" />
        <style type="text/css">
        #centro_wide li { padding-top:10px; }
        </style>
</head>
<body>
<h1>Di&aacute;rio da Rep&uacute;blica Electr&oacute;nico</h1>
<div id="cabeca">
        <div id="quick_links">
                <ul>
                <li><a href="/comum/html/help/ajuda.html" title="Ajuda Geral">Ajuda</a></li>
                <li style="background:none;"><a href="/comum/html/contactos.html" title="Contactos">Contactos</a></li>
                </ul>
        </div>
        <div id="data"> </div>
        <div id="counters"> </div>
</div>
```

Figure 4.8 - Extract from a HTML document

HTML page in TXT format can have hundreds or thousands of lines and represent all HTML formats and text within a web page. Therefore, target information is present in these documents. The key element of wrapping information is to identify simple elements that indicate that a certain string contains the desired information and then cleaning the string to obtain the desired set of data.

In the study phase, the key elements are identified and the programmed will “know” the key words that indicate the presence of the target data.

In some cases information is presented in tables. The HTML tags for tables may miss keywords which identify the presence of the information. To solve this, in this studying phase, the structure of the tables has to be understood and the algorithm has to check the presence of tree structures. Controlling the order of opening tags “<t>” and closing tags “</t>” may be necessary to guarantee that the position of the data has been correctly reached.

Wrapping from other document formats

When the target information is hold in documents, the wrapping procedures can also be done programmatically.

In studying phase it is important to preview solutions for a two-step procedure:

- Converting the collected format to a readable format.

- Construct and algorithm that correctly gathers the information.

The software should have converting capabilities usually to turn any format to a text format such as TXT. In our case studies two examples of these procedures can be reported.

Studying examples of target documents is necessary for a previous understanding of how information is structure within the documents. A testing algorithm should be use over some documents to preview the effectiveness and to prevent possible exceptions.

4.2.3 Crawler programming

We defined crawling procedures as the operation of interpreting the target web site URL and HTML structure to correctly access the target web page or documents from where information is to be collected. The crawling part of the software finishes where Wrapping procedures begin. Crawling may not be a simple process since it depends on the following factors:

- Target website base URL.
- URL structure and handling.
- Information publishing periodicity.
- Website server response to errors.
- String checking.
- Presence of log in and search procedures.
- HTML information processing for other URL constructing.
- Server data base query.
- JavaScript procedures.
- Extracted string encoding.

Since every website can vary in each factor listed above, the crawler capabilities can vary greatly. The crawler must be set according to the target page way of structuring HTML pages or trough URL based queries to the data base stored on the server. Although there are available crawler software's we chose to program our own crawler specifically design for our case studies. The one project crawler design may yield great efficiency and crawling procedures can turn out to be simple.

The crawling structure may also depend on the periodicity of information publishing. In the first case study presented, the information was gathered in a daily basis and the URL was structured accordingly.

In our case studies, the developed crawler first generates the target URL for the corresponding day from which compiles a set of URLs of documents to be downloaded. In other cases, crawling may only generate a URL from which it tests the presence of the desired information. If affirmative, HTML based procedures are triggered for information wrapping and storage.

In a process point of view, crawling is an algorithm that search target URL from which information will be wrapped. Has inputs, the process can be feed with the base URL, variables such as day, year, and month, keywords to be searched, and other variables to align the crawler behavior.

The output of the crawling activity is a list of URLs that represent either web pages or documents to be downloaded, such as PDF's, containing desired information. These URLs can be saved in a file for future use or saved internally in the program.

In the next pages we will describe some challenges encountered in our case studies and other that can be present in crawling activities.

First level URL crawler

A URL crawler used in our first case study was designed according to the used URL structure. Information was organized by an index of days in a year. As an example the following URL has the typical structure:

```
http://www.example.com/news/newsoftheday/getnewsfromday=102.2011
```

The red and green number clearly identified that the requested web page referred to the day 102 from the year 2011. It is then possible to loop through the days from 1 to 365 to gather all information for a given year.

However, in the case studies, the software would already gather all past information, being only interested in recent data. The strategy was to check if there was new information available repeating the last successful attempt and stopping when new information wasn't found. For example, if today's corresponding number was 102, the program would first request number 101 where he would get yesterday's information. Next it would attempt day 102. If new information wasn't found the program would stop. If new information was detected he would start wrapping procedures for information

treatment. Finally the program would request day 103 where he wouldn't find any information and would stop.

The last successful attempt was always register for the future runs. Requesting the previous number guarantees that if late information was added between runs, it wouldn't fail to retrieve them. The following diagram represents the software behavior in crawling activities:

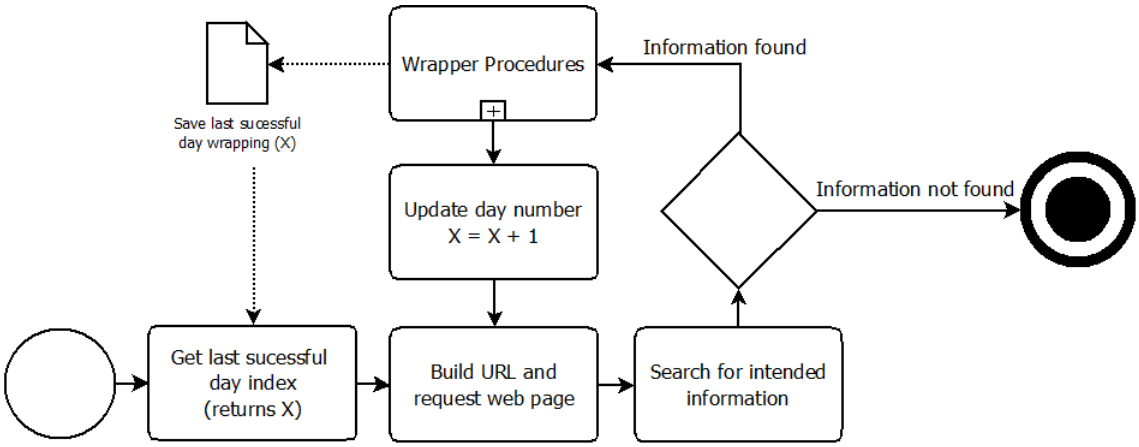


Figure 4.9 - BPMN example of crawler algorithm

Second level URL

Sometimes the target information is not directly present in the HTML text retrieved in the first level URL. For instance, the first level HTML page only retrieves other links where information is to be found. In this case, more treatment to the HTML page has to be done to retrieve more URLs.

We explored two procedures to fetch the second level URL. In this case, those URLs correspond to documents (PDF) to be downloaded later.

The first option procedure is to download the first level HTML page in a text format and search line by line for key elements that represent an URL and at the same time, test if the URL has the intended structure. For instance the target URL has the following structure inside the HTML text file:

```
<p><a href="/util/getpdf.asp?s=udrcp&amp;serie=2&amp;data=2012-06-27&amp;iddr=123&amp;iddip=406209285" title="new item number 27584">Get PDF file</a></p></li>
```

Some elements can be recognized has HTML tags. The key element is the tag “<a>”, seen here in the sub string “<a href”, which indicates the presence of a hyperlink. But, from the page there are hundreds of hyperlinks so the program has to clearly distinct the target links from all the others. In this case, we would test the presence of a substring that objectively identifies a common element present in all target links; we would test the presence of the substring “/collection/getpdf”.

Searching the whole document for this type of structure line by line and cleaning the unwanted elements outputs a list of URLs. Depending of the software structure the output URLs wait for future procedures, therefor they can be save in an output TXT file.

The second method consists in asking to the web control to retrieve all elements for a given tag. In this case a function filtering the elements from the tag type “<a>”.This function responds with an array of all links contained in the document. Next, for each element in the array we would test the presence of the sub string “/collection/getpdf” where, if positive, it was saved.

The same output of an URL list is expected from the two methods. The next list represents an output example from our first case study:

```
/util/getpdf.asp?s=udrcp&serie=2&data=2012-06-27&iddr=123&iddip=406209941
/util/getpdf.asp?s=udrcp&serie=2&data=2012-06-27&iddr=123&iddip=406207332
/util/getpdf.asp?s=udrcp&serie=2&data=2012-06-27&iddr=123&iddip=406206069
/util/getpdf.asp?s=udrcp&serie=2&data=2012-06-27&iddr=123&iddip=406209617
```

However these URLs are not ready for processing since their encoding may not be correct, the root element is missing, and may only be a formal representation of a data base query.

URL encoding and completion

Encoding refers to binary representation of characters. Each program or file format has their specific encoding. Internet browsers deal with HTML file type and URL encoding. A URL text representation in a text format such as HTML may not correspond to the URL representation in the web browser search box. In some cases, information travels different data bases and is written in different file types each with a specific encoding, such as the XML file formats. The two main differences between basic file formats encoding and URL encoding are the characters space and “&”.

An encoding function can be used to normalize and correct these two characters. For a XML file comparison, the following table represents how characters should be replaced:

Table 4.1 -- Characters correspondence between XML and URL encodings

Char.	XML	URL
Space		%20
&	amp&	&

Before requesting a webpage, the extracted URLs encoding has to be checked and corrected if necessary.

Another necessary action is the string completion. An URL representation of a HTML link misses its rout element, in other words, the parent directory in the server. For each URL it was necessary to insert the root element "http://dre.pt" and ensure that the URL was correctly encoded. Finally we have to point out that this URL scheme is in fact a query to the data base to retrieve the document and the real document's URL may be different.

For example, a previous second level URL gather from a first level URL web page comes in the raw format of:

```
/util/getpdf.asp?s=udrcp&serie=2&data=2012-06-
27&iddr=123&iddip=406207332
```

This string is completed and the encoding checked which results in the following:

```
http://dre.pt/util/getpdf.asp?s=udrcp&serie=2&data=2012-06-
27&iddr=123&iddip=406207332
```

The server responds with the next URL and downloads the PDF document:

```
http://dre.pt/pdfgratiscp/2012/06/123/406207332.pdf
```

A Web Client File Download method is used to choose the output directory where the document will be saved.

Automatic login, search and other procedures

In some crawling procedures, before getting to a desired web page, login or search procedures may be necessary. These procedures can be done programmatically by the crawler.

Login procedures can be done assigning a value to two different elements in a web document handled by the web browser control. Traditionally these elements are the username/email and password. The name of these elements must be identified by the HTML element name before programming in the studying phase. A web browser control method allows assigning a value to the correspondent element. After the value have been assign also the submit button element name has to be identified to call a method to click it. Search procedures are alike. Previously the search box element and the search button element have to be identified. In code, the search box value is assigned with keywords and a method to click the button is called.

After one of the procedures a new HTML page is downloaded in response to the login or search procedures from which other crawling procedures can continue.

Another procedure often encounter in crawling is the presence of multiple result pages. For example, when a user search for a keyword in a search engine, a first page will be showed but other pages are available with results. Crawling can call wrapping procedures to gather information and then access the next page. Depending on the web sites structure two methods may be available. The first, always present, is using a button click procedure. The button element name for the next page can be identified previously. Inside crawling procedures, the program will check for the presence of the "next" button. If exist performs the click method. Again, it can wrap the intended information and continue to the next page.

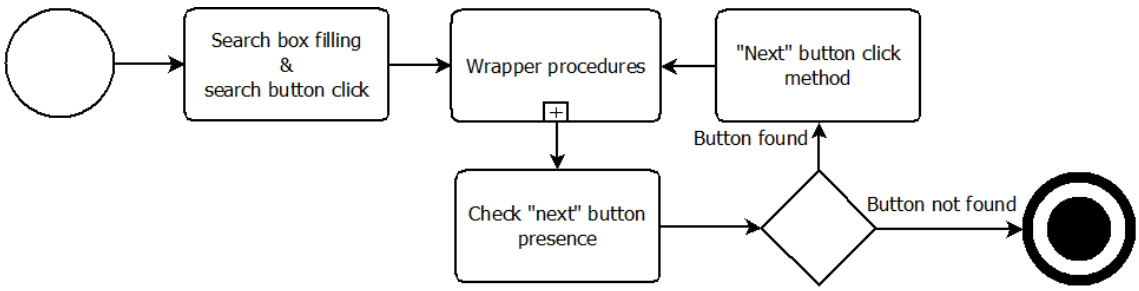


Figure 4.10 - BPMN representation of multiple result pages crawler algorithm

The second method depends on how the web page organizes search results URLs. Observing the URL it may be found a structure similar to:

```
http://www.examplepage/search/pageresults/1
```

After information treatment on this first page, the program will request page number two:

```
http://www.examplepage/search/pageresults/2
```

The program can then check if encounter a “page not found error” or the page was redirected to another URL. If not, wraps the new information and continues to the next result page. A slightly different process is proposed:

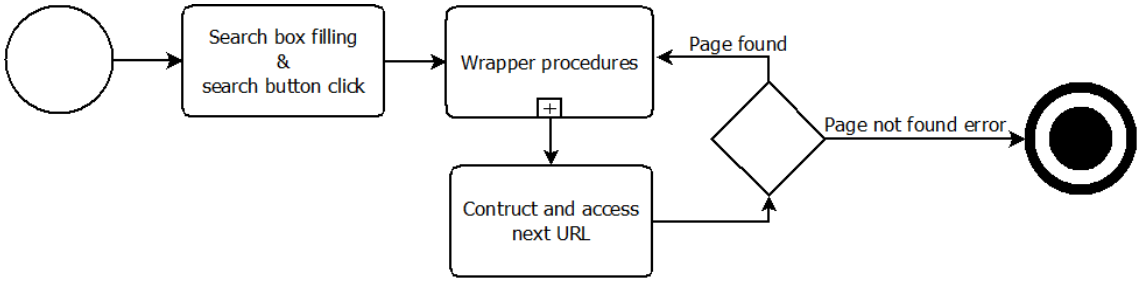


Figure 4.11 - BPMN representation of alternative multiple result pages crawler algorithm

These examples cover key aspects of crawling procedures. Each case may need the combination of different procedures and testing is necessary to ensure that all scenarios are identified and the algorithm responses accordingly.

4.2.4 Wrapper programming

Wrapper programing performs the key element of collecting strings containing the target data. Those strings may represent entity names or values.

Firstly, the output file has to be readable. If not, converting methods have to be designed to transform it to a readable format. Second, the file has to be read and the target information has to be collected. Here we can separate unstructured file formats from structure ones. Unstructured formats can be plain text and a virtual structure has to visualize allowing the algorithm to locate information.

Structure formats can have methods that automatically locate information. In this case, wrapping procedures became simpler and more efficient.

Three possible scenarios may occur depending on the crawler output, its formats and the methods chosen to reach information.

- The basic scenario is to read the files directly. Some file formats can be directly read and no format conversion is necessary such as TXT, HTML, XML, CSV and other Excel formats.
- Other types, like PDF, have to be transform into readable formats or a specific add-ins has to be used to read them. Complete Excel or Word formats have to be managed to extract the text content. XML or TXT are possible outputs of these file conversions.
- A third case can take place. Some directly readable formats may be queried. XML and HTML formats are organized with tags and then can be used to query content. For XML files, the schema has to be known whereas in HTML, tags or element names are necessary.

Wrapping procedures also depend on the amount of data and data types present and if information is to be collected from one or more sources. Hence, before deciding if conversion procedures will be used, a careful study over the file formats and information structure has to be performed to design the most efficient and robust algorithm to reach the data.

In this chapter we will cover a procedure of file converting to a readable format. Then will expose procedures for reading and collecting information from structure and unstructured formats.

Converting file formats

In our first case study the crawler output was a group of PDF files, each one representing an instance of a public tender for the current date. It was used the library itextsharp.dll to read text from the files. Using methods from the library, a simple algorithm was capable of outputting a TXT file with all text present in the PDF. The TXT output file is and unstructured file format.

A commonly expected format from internet data sets is Excel. Since both Visual Basic Express and Microsoft Office are Microsoft applications, internal references to Excel allow a simple conversion procedure. It consist in creating an object instance of an excel document. Using the directory of the crawler output file, an Excel method is used to load the file to the created object. Then

another method allows saving it with another file format such as CVS. CVS file can be later interpreted as a semi-structure file format.

In reality the software is using the actual Excel software to convert the file in an autonomous way. One constrain may arise if this procedure is necessary. It has to be guarantee that Excel is installed in the machine or the methods will fail to open and save the file.

Wrapping from unstructured formats

Unstructured formats have a simple approach to collect data. Reading the file from the first line to the end and checking for the keywords identified in the study phase. The presence of those keywords may signal that the next string, line or value is the target data.

The keyword search method can become complex if a set of variations are necessary to cover all possible scenarios. In our first case study, grammatical errors lead us to check multiple variations of equivalent test strings. During past data collection, missing data was automatically report. It was necessary to track down the documents were the procedure had failed to understand and add up non predicated variations.

It is possible that unstructured formats have a rigid data organization that eases the data collection. CVS files have information separated by ";" characters. If the same order is always present, there's no need to search for keywords. Instead, the data order is known and collected automatically according to its position. Because data is well and consistently presented, this formats may be qualified has semi-structure formats.

Wrapping from structure formats

Structure formats implies that information have an organizational scheme that allows the computer to automatically infer data position.

For instance, XML and HTML files display information according to tags. Those tags identified sets of information in a way that the program can query then.

For HTML formats, and HTML object can handle information and elements name and value. Using specific methods, the program can ask for the value of some HTML elements, interpret table structures, and get all links and other operations. It can first request all text and later handle has an unstructured format.

On the other hand, XML formats have a base structure and nomenclature. The Document Object Model (DOM) has to be studied previously to request information correctly. The library used

to handle XML/RDF formats, have also query capabilities. The algorithm may use the DOM structure to construct a set of queries to collect all information.

Output data consistency check

There's always a relation between data types and the expected formats of the output. When searching for organization/product names that can vary in the number of words, is difficult to programmatically check information consistency. In other data types, however, string formats can be check to ensure that the target information was successfully reached.

Data types such as prices, ID types, dates, can have a format control. The algorithm itself should control the number of characters, some string elements position (such as commas) and the presence of non-numeric characters, to ensure that the correct information was collected.

These procedures help increasing the software reliability and may support error checking and correction. In study phase, expected data formats should be identify and a set of rules should be idealized to program the algorithm consistency checks.

4.2.5 Ontology population

Ontology population defines how gathered information will be introduced in the ontology file. Automatic procedures should be capable of dealing with the wrapper output and compile it in the ontology.

To handle the ontology file programmatically a library of VB functions is used. To do so a reference to that library has to be set. The library is called dotNetRDF.dll and allows defining triple store variables and reason or query over them. The ontology file can be load up to a variable and write back to the same file after information population.

To add information to the ontology a set of base URI and ontology specific URI have to initialize. Base URI are standards ones and are used to assert that some resource is an individual, class, restriction, property and so on. Ontology specific URI have the chosen classes, relations and data type's names. To add information to the ontology, a triple has to be created. Then, three arguments are assigned to the triple including the resource, the relation and the object. That triple is then asserted to the triple store variable in which the ontology has been load to.

After all information assertions, the variable can be written back to the ontology file. Usually this procedure is done in the end of each run.

Our ontology file has a so called RDF/XML format. During development some problems arise when combining schema file with files containing individuals. As a solution we propose that a file with classes and relations (schema) is kept separated from files with information. Before conducting queries or reasoning, both files can be load to a triple store variable. In doing so, all information is combined in a fully capable ontology. Another reason to do so is information size. Ontologies can become heavy if large amount of classes, relations are defined and a great number of individuals instance are created. Manipulation of large ontology files can be slow and lower the software efficiency. To mitigate processing time problems, multiple instance files can be created according to time parameters. For instance, in our second case study, the program keeps one ontology file per month. Before reasoning or querying instance, the needed ontologies are loaded together with the schema file.

To populate ontologies with past information internal variables are forced to take old dates values. If information is kept on the web site the software should be able to collect the corresponded data and populated in the ontology.

4.2.6 Data analysis

The key element of data analysis is to plan relevant data configurations that represent a support for decision making.

According to the CI objectives planned in the first step, data analysis can be program. They should be design in a way that answer strategic questions and support market and competitors characterization. Data analysis uses queried information from the ontology file and further processing. Data is collected according to different parameters or logical relations in order to capture pertinent intelligence.

A user interface can be design not only to visualize analysis results but to search and query specific information. Filters should be created to limit the scope and amount of data. Often, ontology classes are a clear suggestion to relevant filters. For instance, in the first case study the business area identifier (CPV) was very important to hide unwanted instances for other business areas.

In our case studies, however, we only exported information to an excel spreadsheet to demonstrate possible analysis. This proof of concept was necessary since time constrains would not allow full development of a user interface and correspondent analysis procedures.

4.3 Software management

The developed software is in prototype phase and was not deployed in any server or installed in a machine. Therefore, it was not experience continuous software performance. Still some management practices can be idealized in order to prevent miss functions.

One preventive action is to include controlling procedures. The program itself can create simple reports with missed information. Whenever information is missed, a report identifies the instance, date, and related document where the procedure failed. This may advert to possible errors either in the application or target web site side. If critical errors are detected and automatic email can be send to the responsible worker.

To further study problems and if no disk space constrains are present, it is recommend that all original documents are kept in local directories. HTML page files and target documents may be crucial to study errors and design solutions.

Possible events that may cause errors are:

- Changes in the web site HTML.
- Changes in URL structure.
- Grammatical errors in unstructured formats.
- Unexpected exceptions in web server response.
- Changes in test strings.
- Changes in information order in structure formats
- Changes in tag names in markup languages.

These changes can be more or less bypassed with simple changes in the software procedures. Grammatical variations in test strings can be included in code to prevent various forms of data display.

Although it is proposed fully automatized software, periodic inspection to the application is needed to guarantee all functions run as expected.

4.4 Multiple configurations

Multiple configurations is a proposed definition when two software run in parallel and share data to enhance data analysis, add features or complement an existing application.

The only possible concern in this matter is to coordinate read and write procedures in the ontology file between part A and B of the multiple configuration. Internal variables can be used to coordinate ontology file access. For example, while part A is populating ontology, the part B procedures are postponed to the moment part A finished their procedures. Another solution is to share the triple store variable and to assert information from part A and B to the same variable. In this second solution the ontology should be load when the first part initializes and be saved after the last part finishes.

Next we present possible configuration of two software running and sharing information.

Software A works independently and B reads from both ontologies. Read and write coordination is not necessary.

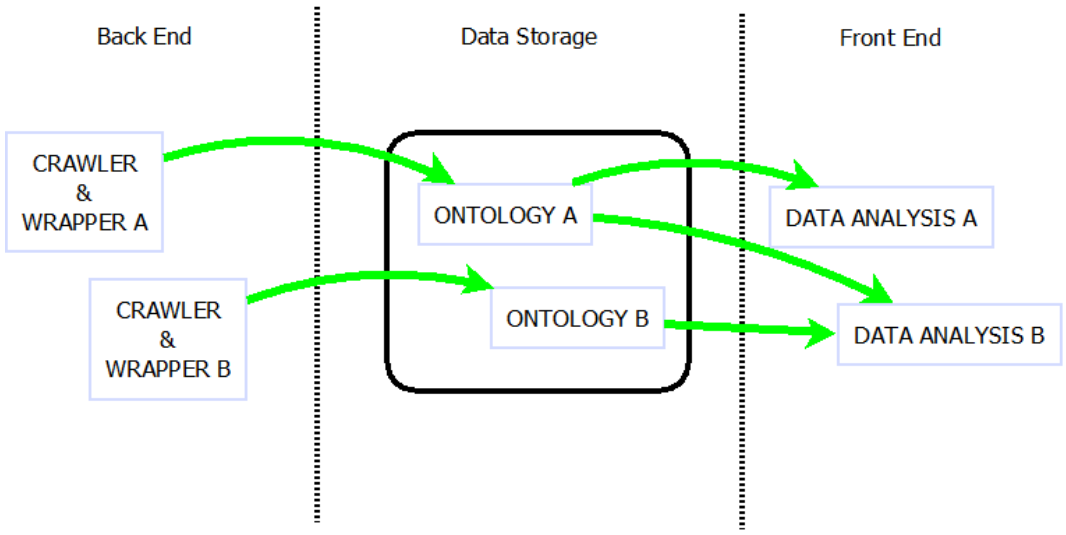


Figure 4.12 - Possible multiple configuration (1)

Software A writes on B's ontology. Read and write coordination is necessary.

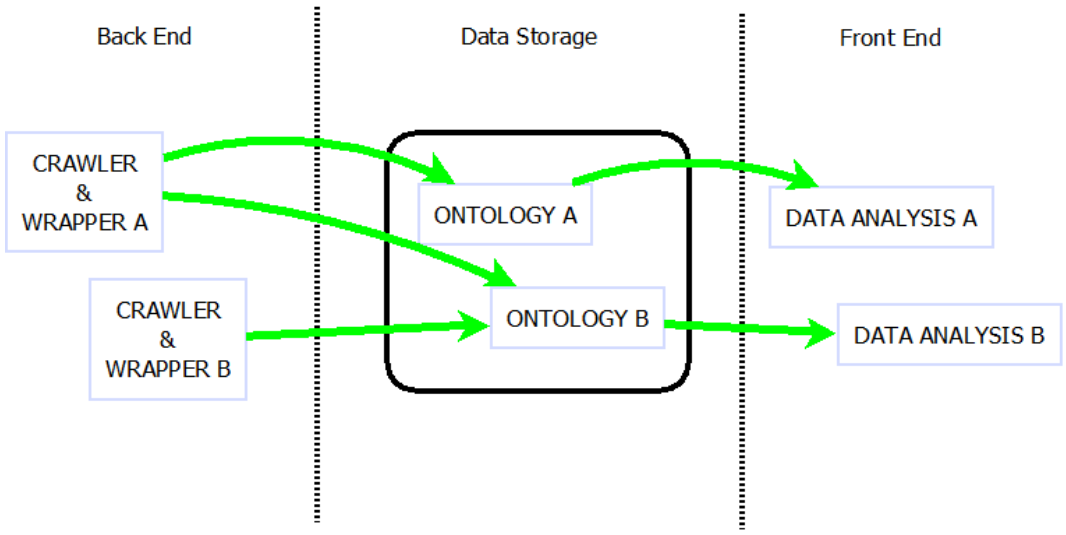


Figure 4.13 - Possible multiple configuration (2)

A third software reads A's and B's ontology. No coordination necessary.

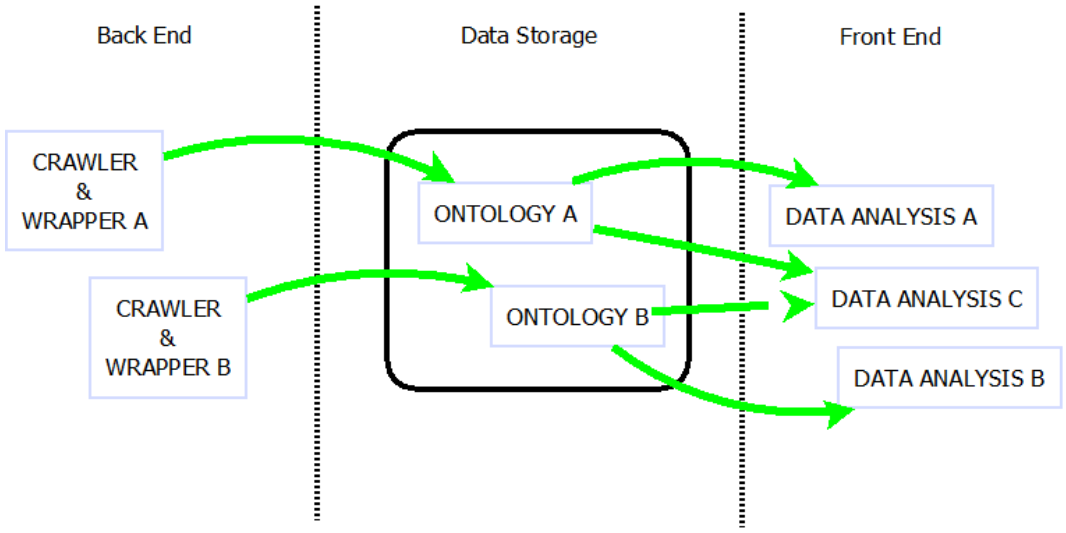


Figure 4.14 - Possible multiple configuration (3)

These configurations may be the result of merging different applications, enhancing the capabilities of an already created one or whatever may arise according to competitive intelligence objectives.

Chapter V – Case studies

In the present chapter it will be presented two cases studies where the methodology was used. The common structure to both cases follows the case study description of the environment, the target objects, software objectives and features.

In both cases it is presented the work flow proposed by the methodology and key aspects that allowed the software to work. As a final element of each case study a conclusion is drawn and the potential of the software is discussed.

5.1 Case study 1 - open public tenders from *Diário da República*

The first case study arises from the need for companies to assess public tenders (PT) they may be interested applying to. In the business competitive intelligence point of view we want to prove that our software could automatically gather web related information and characterize the related business market to support top level decision making.

The target page was dre.pt the official Portuguese Parliament Publication. The developed software had crawler and wrapper capabilities, stored the information in a specifically created ontology and data analysis where made based on 2010, 2011 and 2012 data.

The crawler was able to know the last successful run. It repeats that run and continues to the next day until no more information is found. This prevented that information could be missed if late documents were publish in the previous day.

Wrapping procedures are triggered in between successful crawling activities. The information is gathered, consistency checks are made to ensure no error occurred and each public tender's information was inserted in the ontology.

To gather past information, the software was forced to use variables corresponding to data from 2010, 2011, to May of 2012. After past data collection, the program is capable of autonomously collected up to date information.

From the populated ontology, information from 2010, 2011 and 2012 was massively gathered and data analyses were made. 15432 valid instances were collected.

The results were very interesting since a high level of efficiency was obtaining with minimal fails. For consistency reasons, the software creates reports over all information collection and missed information errors for a continuous improvement of the algorithm's parameters.

5.1.1 Description

General environment presentation

In the Portuguese context, public organizations are obliged to launch a public tender (PT) when they need to acquire products or services that may cost over 75.000€. PTs are also used to call for concessions and qualify suppliers for future contracts. Also private organizations can launch PT if they are interested in finding the best bid for their proposals.

Since 2008 all PTs take place in one of a group of web platforms, here called generally as **Platforms**. In those platforms, organizations place their PTs. These organizations are referred to as **Clients**. Interested organizations apply in those platforms and they are called **Suppliers**. A PT follows a group of procedures that are not relevant in this study which culminates in a contract with the chosen supplier.

The universe of platforms in the Portuguese market is composed by the following players:

- AcinGov
- AnoGov
- ANCP
- ComprasPT
- GateWit
- Saphety
- VortalGov

Each client must choose one platform to publish their PT. Each supplier visits the platform to search for open PT and apply if they choose to.

Another entity comes to play in this case study. By law, every PT has to be published before opening, in a Portuguese Parliament publication. This organization is called *Diário da República* (DRe) and publishes in its official web site all PTs launched at any given day. This website will be

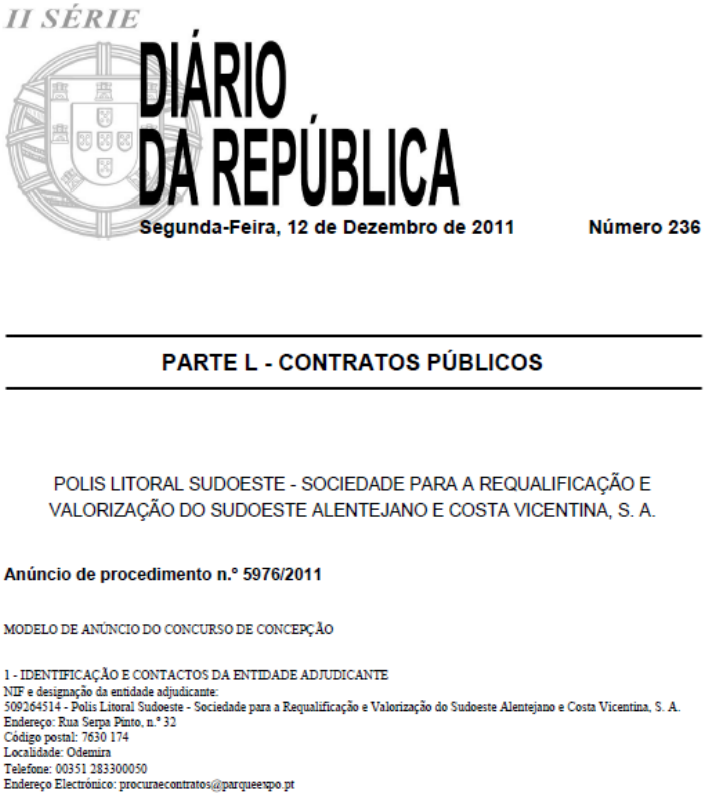
our target page from where information will be collected. This centralized source of information simplified the software since only one web site is necessary for data collection.

There are a few versions of official PT's models that organizations can use from which it was targeted only the standard version. Those versions are:

- PT Standard Model
- PT Rectification Warning
- Due Date Postponement Warning
- System Establishment Model
- Concession PT Model
- Previous Qualification of Suppliers Model

PT Standard Model

From the DRe web page, each PT presented has a PDF file available. A PT standard Model extract is presented in the next figure:



II SÉRIE

DIÁRIO DA REPÚBLICA

Segunda-Feira, 12 de Dezembro de 2011 Número 236

PARTE L - CONTRATOS PÚBLICOS

POLIS LITORAL SUDOESTE - SOCIEDADE PARA A REQUALIFICAÇÃO E VALORIZAÇÃO DO SUDOESTE ALENTEJANO E COSTA VICENTINA, S. A.

Anúncio de procedimento n.º 5976/2011

MODELO DE ANÚNCIO DO CONCURSO DE CONCEPÇÃO

1 - IDENTIFICAÇÃO E CONTACTOS DA ENTIDADE ADJUDICANTE
NIF e designação da entidade adjudicante:
509264514 - Polis Litoral Sudoeste - Sociedade para a Requalificação e Valorização do Sudoeste Alentejano e Costa Vicentina, S. A.
Endereço: Rua Serpa Pinto, n.º 32
Código postal: 7630 174
Localidade: Odemira
Telefone: 00351 283300050
Endereço Electrónico: procuraecontratos@parqueexpo.pt

Figure 5.1 - Extract from a standard public tender document

From this document a set of information has to be presented. The key elements of the PT are:

- Tender ID
- Client
- Business field (given by the CPV)
- Platform
- Geographic location (given by the NUTS code)
- Base Value

The client identifies who is publishing the PT and it is important to suppliers since allows then to know who they are dealing with.

Business field identifies the market the tender refers to. The business field is identified by the *Vocabulário-Comum-para-os-Contratos-Públicos* (CPV) code. As an example, construction companies will only be interested in construction related businesses that are identified with specific CPV numbers.

The platform identifies where the tender is taking place. Companies, when interested in a PT, must be registered in the correspondent platform.

Geographic location is the physical location where the service or product has to be delivered. The location is identified by the European NUTS code protocol. This is important to address if the service/product is in reach and to calculate transportation costs.

Base Value is the maximum value a supplier can offer. Above this value the proposal is invalid. For some companies, only contracts above a certain value are appealing.

Has we will later discuss; these elements are just some of the total information gather by the software.

Competitive intelligence objectives

The objectives must have competitive advantage meaning. How can companies benefit from complete, up to date data about PTs?

Competitive advantage has to come from superior data collection allowing inferring about market trends, competition analysis and general information. Gathering information and reasoning over it would allow a manager to answer the following questions for PT market related activities.

- Is my business area growing?
- What are the main clients in my business area? Should I invest in deepen my relation with them?
- Are my secondary businesses growing? Should I change focus based on that?
- Which percentage of PT I win in my business area?
- Which client is increasing the number of PT they launch?
- What's the average base value of my main clients?
- Is my competition launching PT? What are they acquiring?
- Is my investment in applying to PT paying off?
- Which platform publishes more PT?

Complete data over PT allows a complete analysis to the organization activities in PT related business. This capability is more important as one organization is more dependent on public contracts for survival, such as, construction companies.

General conclusions can also been draw from the whole universe of PT. We can find the top business volume in PT's number and value. We can conclude over tendencies in general business areas, clients, and locations. Every factor: Client, business, platform, location and value, is a matter from which tendencies can be observed, and filters created.

Software features

The software capabilities proposed to fulfill the objectives are:

- To gather all documents from each working day from dre.pt. Each document represents one PT.
- To check the model of the PT and gather information from only the standard ones.
- Correctly collect relevant information from each document and report if errors are encountered.
- To store that information in a suitable ontology.
- Be capable of drawn data analysis based on the collected information from each standard PT.
- Do all previous tasks autonomously with minimal human intervention.

In the next section, it will be presented the key elements in the software development based on our proposed six-step methodology.

5.1.2 Software solution

Ontology building

As stated the first step will be firmly identify what the software competitive intelligence objectives are. In this case the software has the summarized objective of:

- Gather all information from standard PTs in order to best characterize the related market.
- To analyze tendencies over different aspects of a PT. Tendencies over organizations, locations, value, and businesses.

This competitive intelligence should enhance a firm's capability to answer to the proposed questions in the previous chapter and help top level decision making. To accomplish this objective, a serious of parameters from each PT has to be collected. The total amount of information withdrawn from each PT instance is next listed:

- The day number when the PT was published according to the DRe index. Each DRe day number has a date correspondence.
- The PT identifier used by DRe.
- The procedure number of the PT.
- The respective URL from where the PDF can be downloaded from.
- The client who launched the PT.
- The client's NIF.
- The NUT code identifying the geographic location where the product/service has to be delivered.
- The platform which the tender procedures take place.
- The CPV code identifying the business area addressed in the public tender.
- The base value set by the client.

Each PT has a primary ID imported from the DRe classification. The information requirements allow us to construct the following ontology. The full list of ontology elements is presented:

- Classes
  - PublicTender
  - CPV
  - NUTCode
  - Platform
  - Client
  - NIF
- Classes Relations
  - PublicTender -> hasCPVnumber -> CPV
  - PublicTender -> hasNUTCode -> NUTCode
  - PublicTender -> hasPlatform -> Platform
  - PublicTender -> isFromClient -> Client
  - Client -> hasNIF -> NIF
- Data Properties
  - PublicTender -> hasBaseValue
  - PublicTender -> hasDRdayNumber
  - PublicTender -> hasProcedureName
  - PublicTender -> hasURL
  - CPV -> hasDescription

A visual representation of the classes is drawn for easy understanding. Classes are represented with yellow circles, data objects with blue ones. The lines identify the relations orientation of each class and data properties.

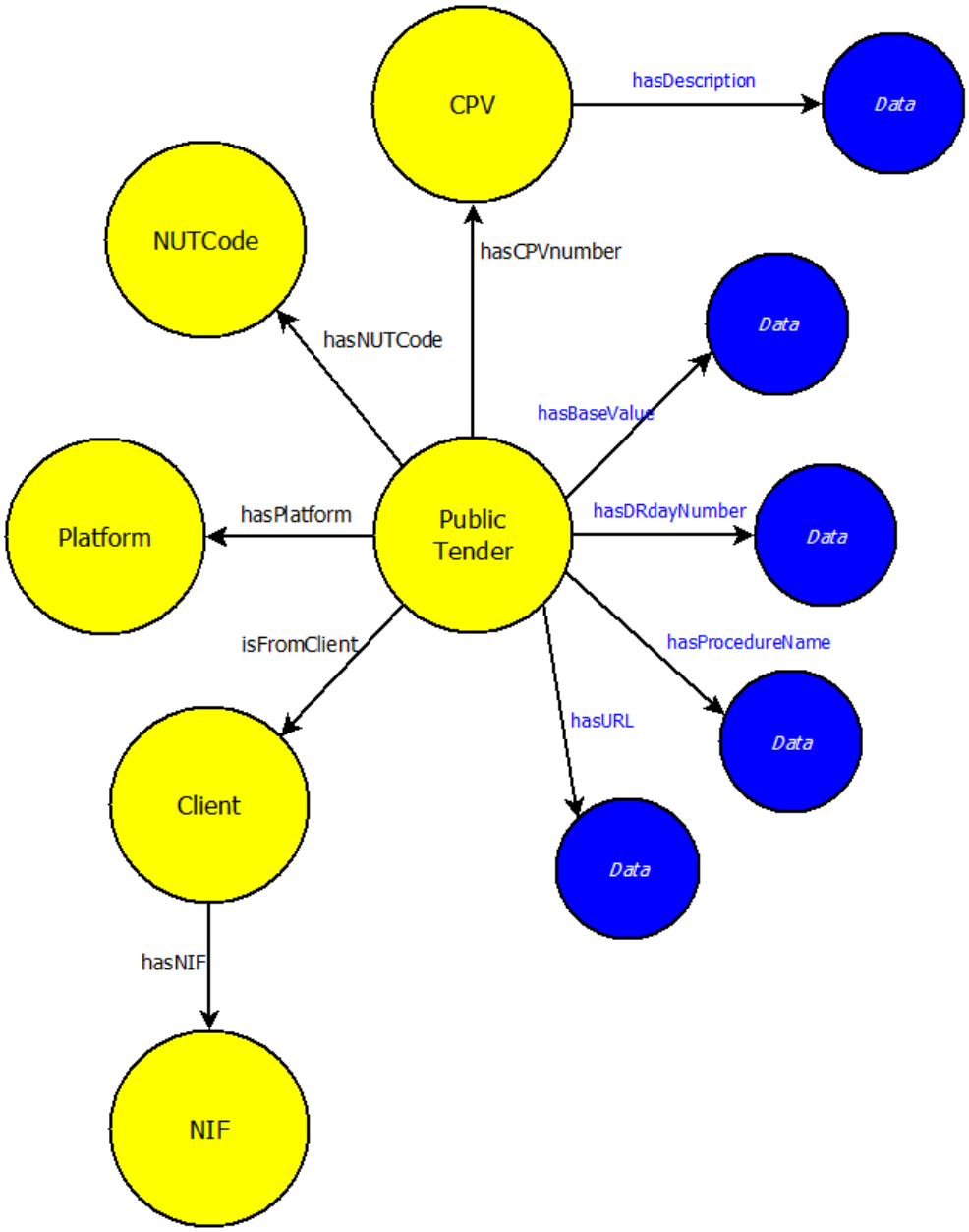


Figure 5.2 - Ontology representation of Case Study I

Web page structure study

Crawling Concerns

As described before a two level URL crawler was developed. For the complete crawler capabilities the DRe web page URL and HTML code were study to identify two key elements.

For the first level URL, the following string was used. The day and year elements are internal variables used to identify the current day.

```
http://dre.pt/sug/2s/getcp.asp?s=udr&iddr105.2012
```

This URL accessed the current page were the public tenders are published. From the HTML code accessed with the previous URL, the following string was used to identify the presence of the PT document's links.

```
/util/getpdf
```

Wrapping Concerns

In order to correctly program the crawler features, some elements had to be studied and determined. A format conversion algorithm was used to extract the all text from each PDF document. Next, for each type of information that had to be gathered, keywords representing the presence of that information were identified. The following substrings were identified:

- To identify that the document was a standard PT model and to get the procedure name. Other elements were used to classify other types of documents.
  - *"Anúncio de procedimento"*
- To check the presence of the CPV number.
  - *"Vocabulário principal:"*
- To get the platform, two test strings were used due to grammatical variations that result from the new grammatical protocol between Portugal and Brazil.
  - *"Plataforma eletrónica utilizada pela entidade adjudicante"*
  - *"Plataforma electrónica utilizada pela entidade adjudicante"*
- To retrieve the NUT code:
  - *"Código NUTS: "*
- To get the base value the next string and some variations of it were used. Other text strings were used to confirm that no base value was presented.
  - *"Valor do preço base do procedimento"*
- The Client and its NIF was gathered after the existence of the next string.
  - *"NIF e designação da entidade adjudicante"*

Crawling programming

The crawling algorithm process is represented in the next figure. Note that wrapping capabilities are trigger within the process.

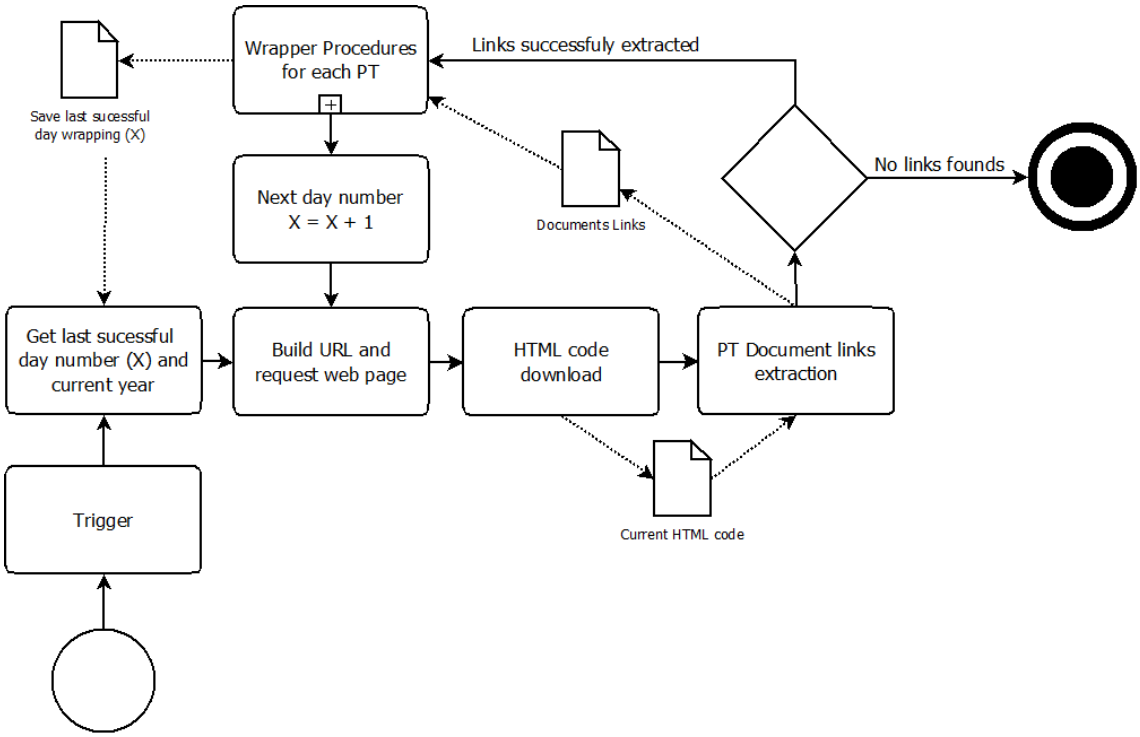


Figure 5.3 - BPMN representation of crawler algorithm of Case Study I

A triggering element is placed in the beginning of the process. During development phase, the triggering action was a button click. For future implementation the triggering event should be a scheduled automated process happening one or more times per day.

Two internal variables are used, the day and year parameters to construct the current URL. The current HTML code, last successful day, and link list documents are saved in local directories to keep the information from run to run even when the software is closed. The crawling procedure also defines when the run finishes.

Wrapper programing

From the previous figure, a task can be exploded to show wrapper procedures. The programing defined a procedure represented by the next diagram:

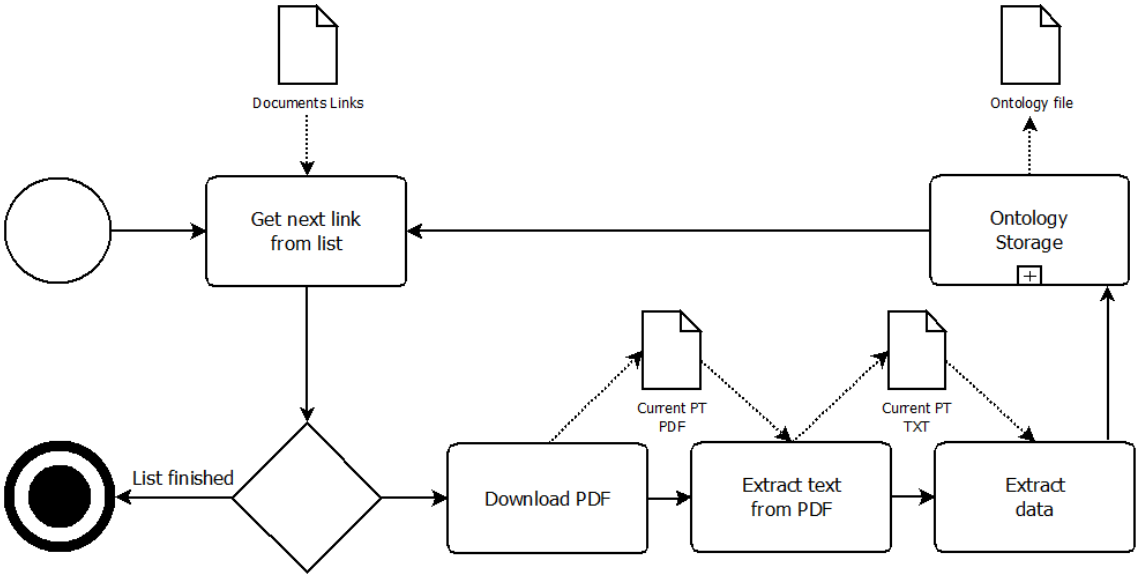


Figure 5.4 - BPMN representation of wrapper algorithm of Case Study I

The text extracting procedure makes use of a Visual Basic code extension called itextsharp.dll. The extension methods allow text extracting to a TXT file which is after organized.

The extract data action is based on a serious of coded methods that reads the TXT files using the test string presented before to gather all PT data. From each link outputted from crawler activities, the wrapper creates an instance for each PT using the native DRe identifier. The identifier is collected from each link and is compose from the last nine numbers in the URL.

```
http://dre.pt/util/getpdf.asp?s=udrcp&serie=2&data=2012-07-
25&iddr=143&iddip=406274709
```

Ontology population

The ontology insertion of data is done using an extension capable of reading and writing the XML/RDF ontology file, the dotNetRDF.dll.

The procedure simply initializes one instance of the main Public Tender classes with the DRe identifier. Associated to the PT instance, it fills all the collected data using triples. For some types of information, some individuals may not be original, such us Clients. In those cases, the algorithm first checks the presence of the individual.

The only information not associated with the PT is the client's NIF. This element is correlated to the Client's name.

Data analysis

Data analysis is only possible when some amount of information is collected. In order to gather significant information, the software was force to use day and year variables that did not correspond to the current date. It was gather all PTs from 2010, 2011, and until May of 2012. All data was extracted from the ontology to a TXT file to be handled in MS Excel. This approach is done only for proof of concept since we propose that data analysis should be integrated within the software and available from the user interface. Further programming was necessary but time constrains didn't allowed to complete that capabilities.

Filters over the various types of information were created and automatic functions respond to user choices with visual information. For example, to visualize CPV related information, the user can choose a CPV number in the Excel spreadsheet. After inputting the CPV main category number a serious of analysis are presented:

Table 5.1 - Public tender data according to a specific CPV category

CPV Category:	55	Serviços de hotelaria, restauração e comércio a retalho			
	Total	**2010**	**2011**	**2012**
PT (#)	582	247	241	94
PT (%)	3,77%	3,79%	3,63%	4,13%
∑ BV	288.561.768,72 €	165.991.578,94 €	68.403.756,54 €	54.166.433,24 €
∑ BV (%)	2,54%	3,18%	1,40%	4,29%
BV Average	495.810,60 €	672.030,68 €	283.833,01 €	576.238,65 €

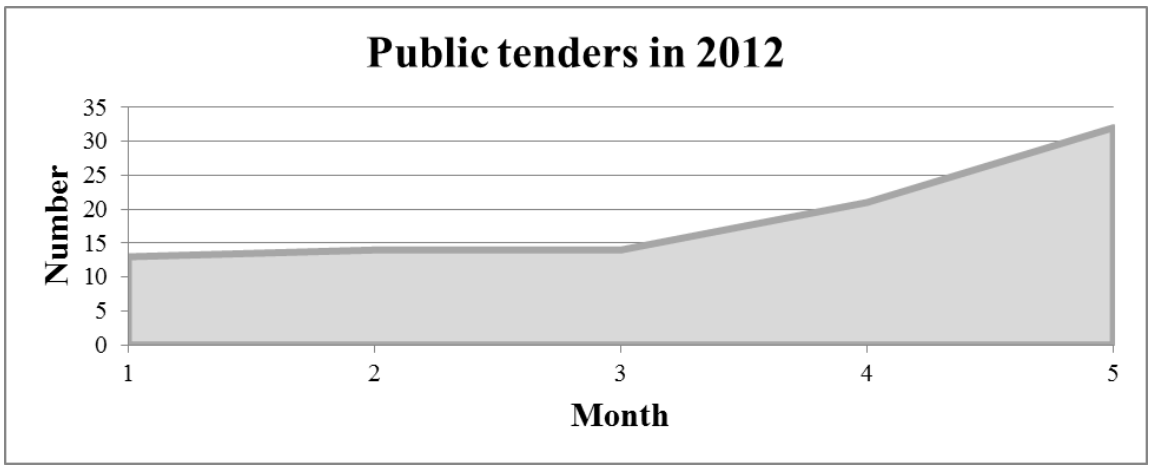


Figure 5.5 – Number of PTs published per month during 2012 for the specified CPV category

Similar analyses were programed according to the client if they launched more than 20 public tenders.

Table 5.2 - Public tenders data according to a specific client

Select client. Only clients with more than 20 PT available.

INSTITUTO NACIONAL DE SAUDE DR. RICARDO JORGE I.P.				
	Total	2010	2011	2012
PT (#)	87	10	50	27
PT (%)	0,56%	0,15%	0,75%	1,19%
∑ BV	8.768.560,67 €	649.737,47 €	5.376.845,98 €	2.741.977,22 €
∑ BV (%)	0,08%	0,01%	0,11%	0,22%
Average BV	100.788,05 €	64.973,75 €	107.536,92 €	101.554,71 €

Table 5.3 - Used CPV categories and NUT codes used by a specific client

Preferred CPV			
	PT (#)	Main Cat.	Description
1º	55	33	Equipamento médico, medicamentos e produtos para cuidados pessoais
2º	17	24	Produtos químicos
3º	7	31	Maquinaria, aparelhagem, equipamento e consumíveis eléctricos; iluminação

Preferred NUT codes		
	PT (#)	Main Cat.
1º	87	PT171
2º	-	-
3º	-	-

Also general analyses are possible:

Table 5.4 - General data concerning all collected public tenders

General Data

	Total	2010	2011	2012
Total number of Public Tenders	15432	6524	6631	2276
Total Base Value	11.362.267.908,37 €	5.212.634.577,82 €	4.887.993.393,11 €	1.261.596.673,93 €
Average Base Value	736.279,67 €	798.993,65 €	737.142,72 €	554.304,34 €
Average public tender per Client	10,97	7,19	6,46	3,49
Active Clients	1407	907	1026	653
Average Base Value per Client	8075528,01	5.747.116,40 €	4.764.126,11 €	1.932.001,03 €
Average Public Tenders per day	25,44	26	27	21

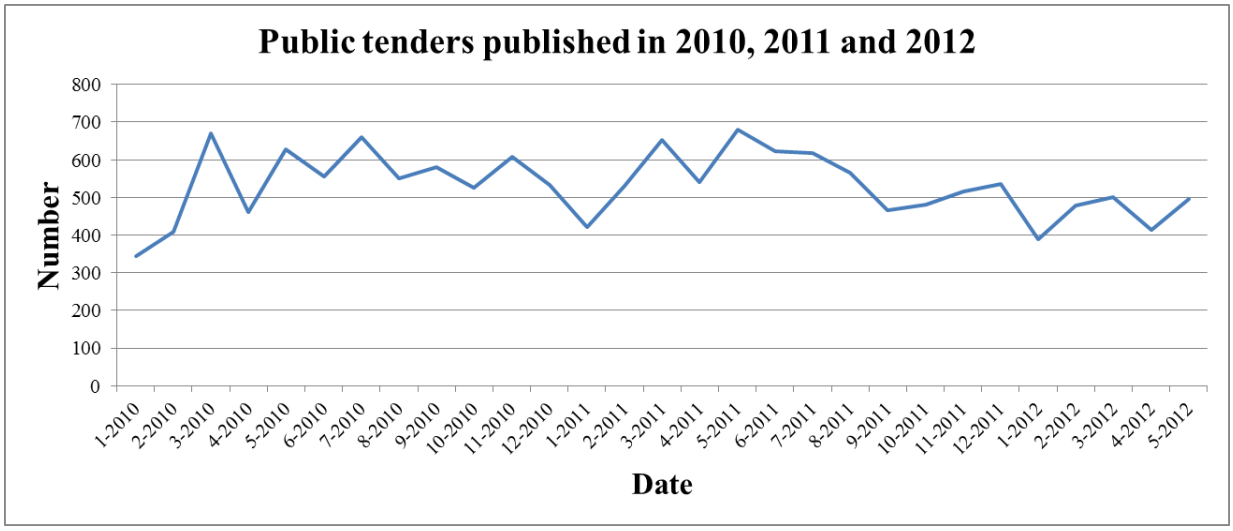


Figure 5.6 – Number of PTs published per month during 2010, 2011 and 2012

These are a sample of possible data filtering and arrangement.

All the presented graphics and tables serve to show that superior conclusions over different aspects of public tender activity can be drawn. Careful and relevant observation of the data should enable top level management to modify public tender proposals policies in a way that best suits the market environment.

This tool should enhance a single firm advantage against other players that are unable to visualize the present data.

5.1.3 Discussion

Objective achievements

The software achieved the objective of collecting all standard PT data efficiently. Collection failure is related to keywords' variations. Those variations can be identified and inserted in code to enhance data collection effectiveness in future runs.

The data analysis is possible and can enhance the organization competitive advantage. Conclusions over specific business areas, clients, location and general environment can support decision making.

Additional features possibilities

The decision of collecting the client's NIF had the objective of supporting other types of data collection that can enhance information value. For example, through the client NIF it is possible to collect public data to characterize financial health, payment behavior, and other parameters that should help a potential supplier to better access the public tender attractiveness. With a simple ontology modification or merging a new ontology, the average payment time can be added to the client information. This information can be attached to the public tender and an attractiveness classification can be programed.

From a functional point of view, some features can be added helping internal procedures efficiency. For companies that regularly search for public tender, some work hours are spend looking for new public tenders each day or each week. This search could be suppressed since the software already collects crucial information from all public tender. The internal process of search could be simplified by simple filtering the ontology data. Another possibility is to program automatic notifications based on pre-determined parameters. For example, when a public tender that respects a minimum base value, from a given business area is detected, a notification is sent via email adverting for an interesting new PT.

Multiple configuration possibilities

If second software is programed to collect data that somehow can correlate with this subject, a multiple configuration is achieved. If this programed worked in parallel, data analysis could feed from both ontologies. That information can enhance analysis value.

An example of multiple configurations would be the integration of the two case studies. This possibility will be discusses in Chapter 5.2 when the second case study has been already presented.

Market value potential

Market value potential has two components: the competitive intelligence value and the functional capabilities.

The advantage withdrawn from gathered intelligence should help a company to better align their policy to the market environment. The software enables a great amount of information to be automatically gathered with minimum human intervention. This information is then filtered, treated and visually organized in order to characterize market behavior, competitors power and evaluated the organization's performance. Using the right information display, strategic decision can be supported by this intelligence.

This strategic position may have different objectives:

- close more contracts,
- to increase average contract price,
- have aggressive pricing with high margin products against competitors,
- increase margins in less aggressive markets,
- displace the main business to secondary products or services.

Since the application is running continuously, the market response to new strategic position can be observed in real time therefore, it can be continuously evaluated. This constant adaptation is a dynamic capability and can improve the organization's survival capability. A smarter investment in public tender submissions should increase competitiveness by winning more public tenders or by winning more valuable ones.

The market potential of the functional capabilities is justifiable with a more efficient internal procedure, decreasing time wasting and diverting some workforce to more valuable tasks. Periodically, companies interested in public tenders spend hours searching new public tenders. This means log-in to dre.pt, reading PT documentation. The application suppresses those tasks since key information of all recent PTs are presented on screen in real time. Searching process can be avoided and time can be used on other added value tasks.

5.2 Case study 2 - closed public contracts from *Base*

The second case study relates to the first one but an independent approach is intended. Here it was gather information about all public contracts, including public tenders. Again the software must have to retrieve all related information in an autonomous manner. Public tenders from private companies are outside reach because only public institutions have to present sealed contracts information.

The target web page is a governmental site called base.gov.pt. The crawler capabilities were able to collect daily data from sealed public contracts. The wrapper features, made sense of the output document to retrieve data and store in a designed ontology.

Similar procedures were adapted from the first case study that showed that a base structure can be encountered for this kind of software. Therefore not only a methodology can be proposed but a solid scheme foundation could be created.

To gather past information, the software allows the introduction of an initial and final date inputs from which it collects all data in-between those dates. Using those input fields, data from January 2012 to July 2012 was collected.

Due to a bigger amount of information, a different ontology approach was developed. Instead of having one central ontology file, each month has a correspondent file. All ontology files share the same base schema.

The results allow us to study public organizations buying behavior. Critical business areas, most active organizations, and spending values can be characterized. Competitor's contracts can be tracked to gain competitive intelligence that allows a comparison between the organization and its competitor's market share.

5.2.1 Description

General environment presentation

Public organizations such as Ministries, Hospitals, Municipalities, and all public institutions have to report information about contracts they closed with another public or private company. This commercial relationship can have different justifications and be made upon different standards. The type of commercial relationship is referred to as the procedure type whether the content is described has the contract type.

The procedure types that have to be reported are:

- Direct contract.
- Public Tender.
- Public Tender with previous supplier qualification.
- Negotiation procedure
- Commercial Dialog

The last two elements are very rare and may not have a price associated with it.

Why the public organization engaged in a commercial transaction has also to be clarified using different contract types categories:

- Products acquisition.
- Services acquisition.
- Concession of a public construction.
- Concession of public services.
- Building contract.
- Products allocation.
- Partnership.
- Other.

The contracts and related details have to be reported to a governmental web page, base.gov.pt which, for obvious reasons, will be the target page.

Closed Contract Parameters

The next figure represents a sealed contract details presented in the web page.

Detalhe do Contrato Imprimir

DATA DE PUBLICAÇÃO NO BASE	06-08-2009
TIPO(S) DE CONTRATO	Empreitadas de obras públicas
TIPO DE PROCEDIMENTO	Concurso público
DESCRIÇÃO	Repavimentação da EN 223 - Corga/ Fagilde - Santa Maria da Feira
FUNDAMENTAÇÃO	Artigo 19.º, alínea b) do Código dos Contratos Públicos
FUNDAMENTAÇÃO DA NECESSIDADE DE RECURSO AO AJUSTE DIRETO (SE APLICÁVEL)	Não Preenchido
ENTIDADE ADJUDICANTE - NOME, NIF	MUNICÍPIO DE SANTA MARIA DA FEIRA (501157280)
ENTIDADE ADJUDICATÁRIA - NOME, NIF	PAVIAZEMÉIS - PAVIMENTAÇÕES DE AZEMÉIS, LDA. (502896604)
OBJETO DO CONTRATO	A presente empreitada compreende o levantamento de tampas e cabeças móveis em ferro fundido, nas redes de saneamento, águas pluviais e abastecimento de água, com nivelamento à cota do pavimento, execução de camada de desgaste em betão betuminoso e execução de sinalização horizontal.
CPV	45230000-8, Construção de condutas de longa distância, de linhas para comunicação e transporte de energia, vias rápidas, estradas, aeródromos e vias férreas; nivelamento
DATA DE CELEBRAÇÃO DO CONTRATO	04-08-2009
PREÇO CONTRATUAL	68.996,69 €
PRAZO DE EXECUÇÃO	45 dias (1 mês e 14 dias)
LOCAL DE EXECUÇÃO - PAÍS, DISTRITO, CONCELHO	Portugal, Aveiro, Santa Maria da Feira
CONCORRENTES	-
ANÚNCIO	-
INCREMENTOS SUPERIORES A 15%	-
DOCUMENTOS	-
OBSERVAÇÕES	-

Figure 5.7 - Public contract details in base.gov.pt

For the environment characterization it is of interest to collect the following information:

- Procedure type
- Contract type
- CPV
- Client
- Client's NIF
- Supplier
- Supplier's NIF
- Price (€)
- Due time (days)

Competitive intelligence objectives

The competitive intelligence advantage withdrawn from the data analysis is similar to the one possible in the previous case study. The significant differences are that the universe of information is more extensive since it covers more contracts type's then only public tenders; the price information is the actual contracts value instead the ceiling limit that a public offer could have.

The detailed data analysis would support decision making from a group of proposed questions from which a company may gain a better insight over the public contracts market:

- In which business areas, public organizations are spending more? Which ones are growing and declining?
- Organization's business areas tendencies are favorable to the company's activity?
- Are my secondary businesses growing? Should I change focus based on that?
- Which organizations are more active? Should I invest in a more close relationship with then?
- Which competitors are engaging in more contracts? Or more valuable ones?
- Could our organization practice better prices in specific contracts?
- Could we redirect efforts to business areas where the organization can offer more competitive prices?
- How can the company prepare to predictable public spending cuts?

Those are strategic question that could fundament re-positioning in the public contract market. Conclusions drawn from analysis should support top level decision making.

Also in this second case study, general information can also be studied.

- How big is my business market in public contracts?
- What's the total value of governmental spending?
- Is the average contract value increasing or decreasing?

Software features

The software capabilities proposed to fulfill competitive intelligence objectives are:

- To gather all information from each public contract.
- Treat semi-structure information in an accurate manner.
- Correctly collect relevant information from each contract and report if errors are encountered.
- To store that information in a suitable ontology.
- Be capable of drawn data analysis based on the collected information.
- Do all previous tasks autonomously with minimal human intervention.

In the next section, it will be presented the key elements in the software development based on our proposed six-step methodology.

Note that a common structure can be observed between case studies.

5.2.2 Software solution

Ontology building

As the first and most important step, the fundamental objective has to be objectified. According to the competitive intelligence objectives proposed, what capabilities the software have to be capable of?

- Gather all information from public contracts in order to best characterize the related market.
- To analyze tendencies over different aspects of a PC. Tendencies over organizations, value, and businesses.

The information withdrawn from a public contract had some restrictions. Firstly, the information available to be retrieved from the target page and second, the data types structure have to be sufficiently structure to be handled. For instance, location information is given but is not structure. Sometimes, only a district name is given, in other cases, complete information with district, town and local names are described. Due to this unstructured form, location data collection would have no value since we it wouldn't be possible to related information. The programming effort to normalize data would outcome the value of it. Based on these factors, the following set of information is gathered from the target page:

- The day the public contract is publish in the web page. From the date, an identifier is constructed to name a PC instance.
- The procedure type.
- The contract type.
- The client organization and respective NIF.
- The supplier and respective NIF.
- The contract price.
- The due time.

The NIF element is very important to identify organizations. Due to names variations, some organization can have a group of equivalent designations. The NIF, however, doesn't change, therefor is more appropriate to identify any organization – clients and suppliers.

Some NIF don't follow the expected format. This is due to the presence of foreign organizations that have a different number format. Other cases happen due to formatting errors from the target page. The NIF data collection procedures should be optimized to correct some error types. However, the software capabilities are not threaten by this element.

According to this information the following ontology structure is proposed:

- Classes:
  - Contract
  - ProcedureType
  - ContractType
  - Client
  - Supplier
  - NIF
  - CPV
- Classes Relations
  - Contract -> hasProcedureType -> ProcedureType
  - Contract -> hasContractType -> ContractType
  - Contract -> hasCpvNumber -> CPV
  - Contract -> isFromClient -> Client
  - Contract -> hasSupplier -> Supplier
  - Client -> hasNIF -> NIF

- Supplier -> hasNIF -> NIF

- Data Properties
  - Contract -> hasContractPrice
  - Contract -> hasDueTime
  - CPV -> hasDescription

A visual representation of the ontologies follows. The same color scheme was used.

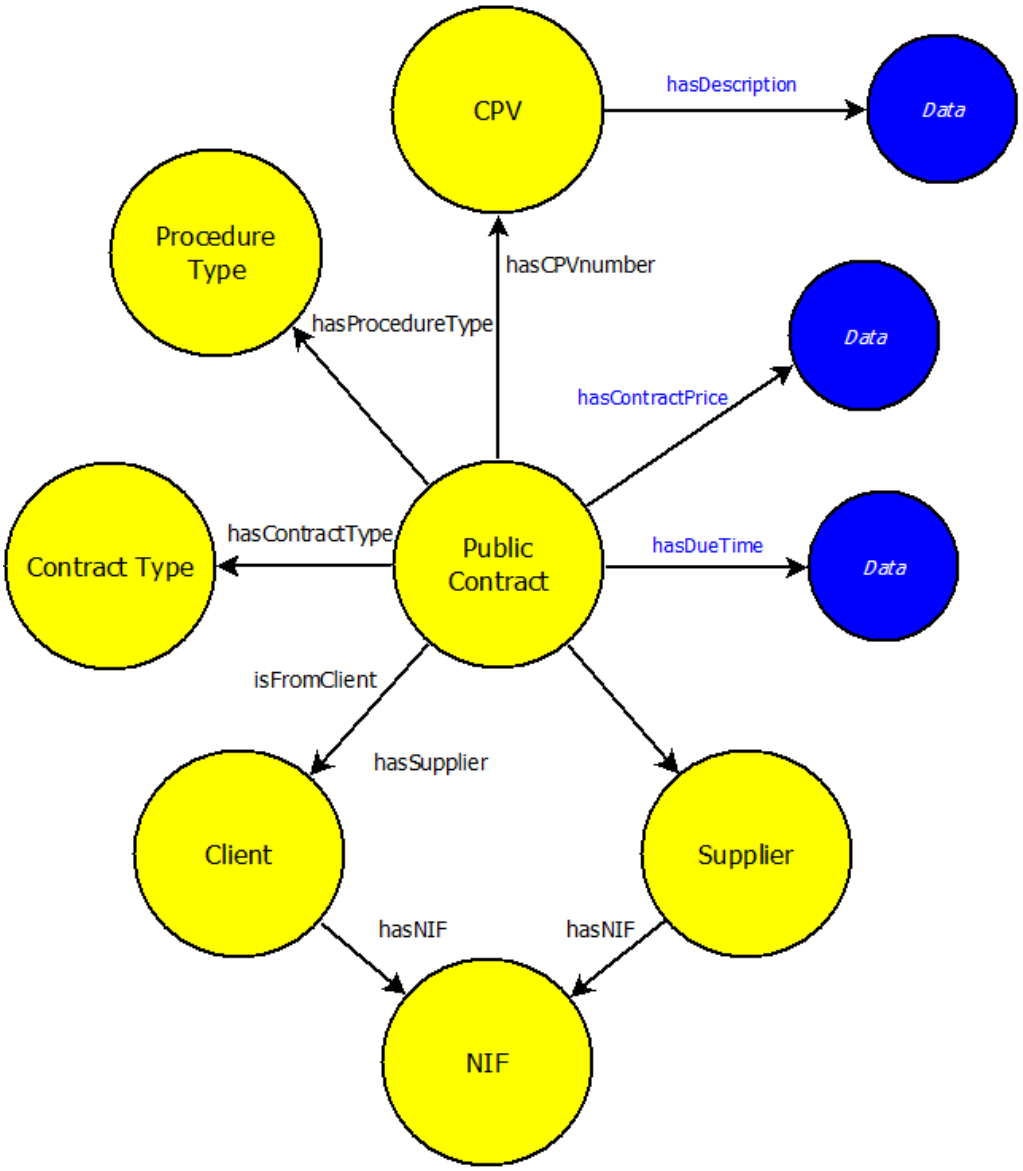


Figure 5.8 - Ontology representation of Case Study II

Web page structure study

Crawling Concerns

A simple crawler procedure was obtained to gather a document with all search results from a given date input. The following base URL was identified using Google Chrome features. The **input** word identifies the date input place.

```
http://www.base.gov.pt/base2/rest/contratos.csv?texto=&tipo=0&tipocontrato=
0&cpv=&adjudicante=&adjudicataria=&desdeprecocontrato=&ateprecocontrato=&de
sdedatacontrato=&atedatacontrato=&desdedatapublicacao=INPUT&atedatapublicac
ao=INPUT&desdeprazoexecucao=&ateprazoexecucao=&pais=0&distrito=0&concelho=0
```

This simple element and the date input handling were sufficient to successfully program crawler capabilities. The crawler output is a CVS file to be handle by the wrapper.

Wrapping Concerns

The crawler file output structure was study to know the information position in each text line. Generally, each result instance from the search was display by line. Each line had all related information, with the same structure, with information separated by the substring element ";".

The study consisted in identifying the data order and consistency to program the algorithm accordingly.

Crawling programming

Next it will be presented the algorithm process diagram. The structure is similar to the one presented in case study one. The program is capable to determine the current date, however, since the web page may not publish all data in the current day, search are made to the previous day.

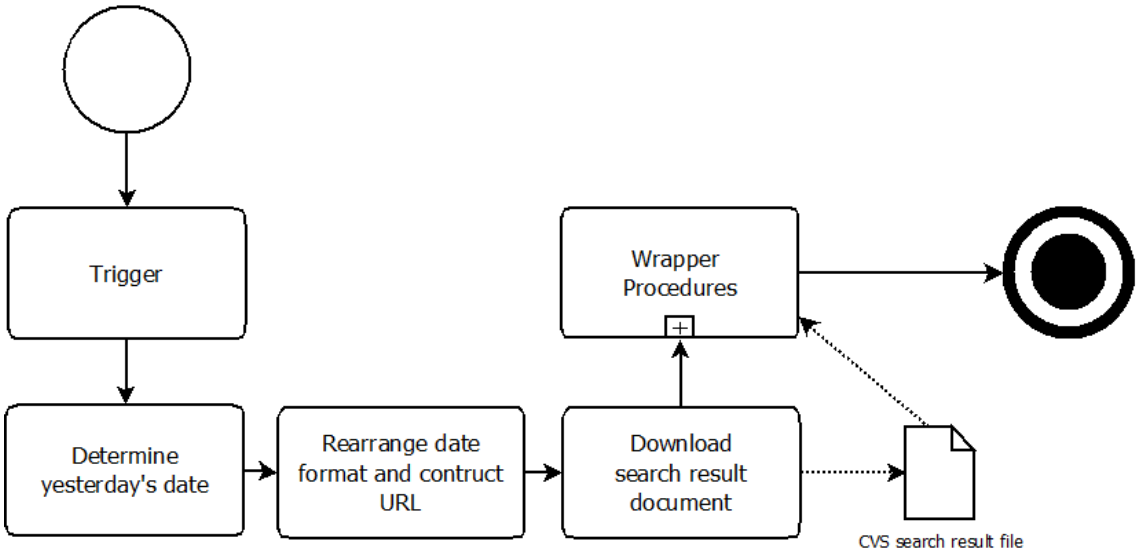


Figure 5.9 - BPMN representation of crawler algorithm of Case Study II

The search procedures focus on the day before the current date. So, it isn't intended to search for failed information and no iterations are necessary, just one simple run. This simplifies the algorithm structure and no exit testing is necessary.

The trigger element should be a scheduled run for a given moment in each day. As describe in case study one, during development phase it was used a button click event.

To easily collect large amount of past data, to input date boxes were created. The program was then forced to gather all data from the beginning to the finish date using the describe process.

Wrapping Programing

The previous wrapping action is exploded to the following process. Since the output file is already structured, the wrapping process is quite simple and is only responsible for the information reading and ontology population.

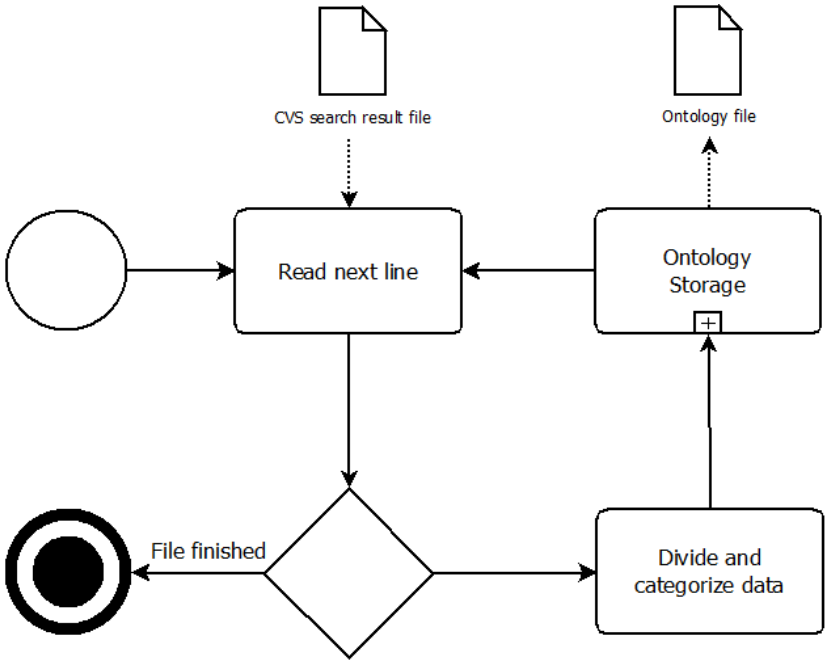


Figure 5.10 - BPMN representation of wrapper algorithm of Case Study II

The file can be read directly and no format conversions are necessary. No intermediary files have to be created and the process becomes unsophisticated.

Since no native identifier of an instance is given, one has to be created. The algorithm used the current date and adds a four digit number corresponding to the line instance. As example, the third PC instance of 10th of July 2012 has the following identifier:

```
201206100003
```

This guarantees that no two instances can have the same identifier and allows the occurrence of a maximum of 9999 instance in one day.

Ontology population

In the second software, also the dotNetRDF.dll extension was used to handle the XML/RDF ontology file.

The procedure simply initializes one instance of a Public Contract class with the created identifier. Associated to the PC instance, it fills all the collected data using triples. For some types of

information, some individuals in some classes may not be original, such us Clients. In this case, the algorithm first checks the presence of the individual.

The information insertion follows the ontology structure presented in figure 5.8. The only information not associated with the PT is the client's and supplier's NIF.

Data analysis tool

To have significant data to design data analysis, information from 2012 was collected. The software was forced to gather information from January to July. All data was exported to an MS Excel spreadsheet where data analyses were drawn. This is just for proof of concept since we intend to integrate data analysis capabilities within the software.

Filters over the various types of information were created and automatic functions respond to user choices with visual information.

The user can gather data from any Client or Supplier. Information is organized by total number of contracts which are then discriminated per contract type:

Table 5.5 - Public contracts' data according to a specific client

			2012					
		Total	01	02	03	04	05	06
All Contracts	#	10	3	2	3	0	1	1
	Total Value	262.227,20 €	152.542,68 €	14.963,60 €	73.420,92 €	0,00 €	7.500,00 €	13.800,00 €
	Average Value	26.222,72 €	50.847,56 €	7.481,80 €	24.473,64 €	-	7.500,00 €	13.800,00 €
Direct Contracts	#	8	2	2	2	0	1	1
	Total Value	130.984,52 €	85.200,00 €	14.963,60 €	9.520,92 €	0,00 €	7.500,00 €	13.800,00 €
	Average Value	16.373,07 €	42.600,00 €	7.481,80 €	4.760,46 €	-	7.500,00 €	13.800,00 €
Public Tenders	#	2	1	0	1	0	0	0
	Total Value	131.242,68 €	67.342,68 €	0,00 €	63.900,00 €	0,00 €	0,00 €	0,00 €
	Average Value	65.621,34 €	67.342,68 €	-	63.900,00 €	-	-	-
With Previous Qualif.	#	0	0	0	0	0	0	0
	Total Value	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €
	Average Value	-	-	-	-	-	-	-
Others	#	0	0	0	0	0	0	0
	Total Value	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €
	Average Value	-	-	-	-	-	-	-

The top three business areas are displayed.

Table 5.6 - Used CPV in public contracts for a specific client

	Preferred CPV		
	PC	CPV	Description
1º	3	79	Serviços a empresas: direito, comercialização, consultoria, recrutamento, impressão e segurança
2º	2	50	Serviços de reparação e manutenção
3º	2	48	Pacotes de software e sistemas de informação

A spreadsheet displays the top 20 clients and suppliers per contract number and value.

Table 5.7 - Top suppliers according to contracts total value

Position	NIF	Client	Contracts	Total Value
1	500197814	MOTAENGIL ENGENHARIA E CONSTRUCAO SA	26	143.247.828,89 €
2	500097488	TEIXEIRA DUARTEENGENHARIA E CONSTRUCOES LDA	14	83.675.378,16 €
3	500553408	ALEXANDRE BARBOSA BORGES SOCIEDADE ANONIMA	18	70.649.768,47 €
4	500073791	CONSTRUTORA ABRANTINA SA	7	68.553.595,61 €
5	500073880	LENA ENGENHARIA E CONSTRUCOES SA	7	64.732.112,44 €
6	500207577	OPWAY ENGENHARIA SA	4	62.082.902,01 €
7	502314311	PATRICIOS SA	4	57.038.393,27 €
8	500195838	) ALGARVE SA 500201145 MSF MONIZ DA MAIA SERRA E FOI	2	47.925.567,00 €
9	500265445	HAGEN ENGENHARIA SA	2	47.633.098,52 €
10	501176454	J GOMES SOCIEDADE DE CONSTRUCOES DO CAVADO SA	3	46.180.477,53 €
11	500090114	ARDINAGEM LDA 502994614 EDIFERCONSTRUCOES PIRES C(	4	44.658.382,13 €
12	501112308	IEL AS COUTO LDA 500072868 MONTEADRIANO ENGENHARI	5	42.807.293,22 €
13	505924170	IE CONSTRUCOES SA 502273941 SOCIEDADE DE CONSTRUCC	1	42.391.872,67 €
14	500285608	TOMAS DE OLIVEIRA EMPREITEIROS SA	6	41.153.050,65 €
15	500739749	73791 LENA ENGENHARIA E CONSTRUCOES SA 500073880 M	2	40.792.530,22 €
16	509702317	LOTE 3N15 EDIFER ENSULMECI ACE	1	40.477.747,00 €
17	503156000	)285608 NEOPUL SOCIEDADE DE ESTUDOS E CONSTRUCOES	3	40.412.942,78 €
18	980048095	CONSTRUCTORA SAN JOSE SA	6	40.224.146,89 €
19	503504564	EDP COMERCIAL COMERCIALIZACAO DE ENERGIA SA	69	38.964.903,20 €
20	500018936	ALVES RIBEIRO SA	2	38.804.833,78 €

General information is also characterized:

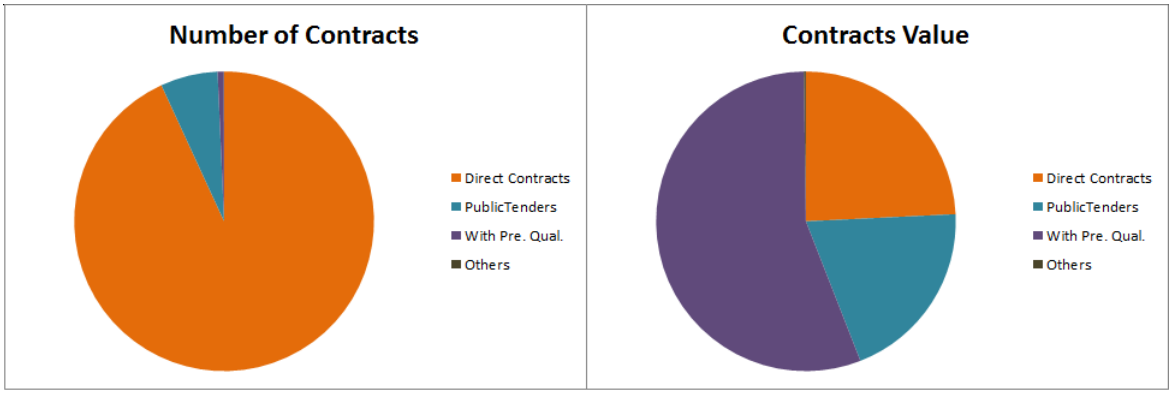


Figure 5.11 – Number of contracts and contracts' value per procedure type.

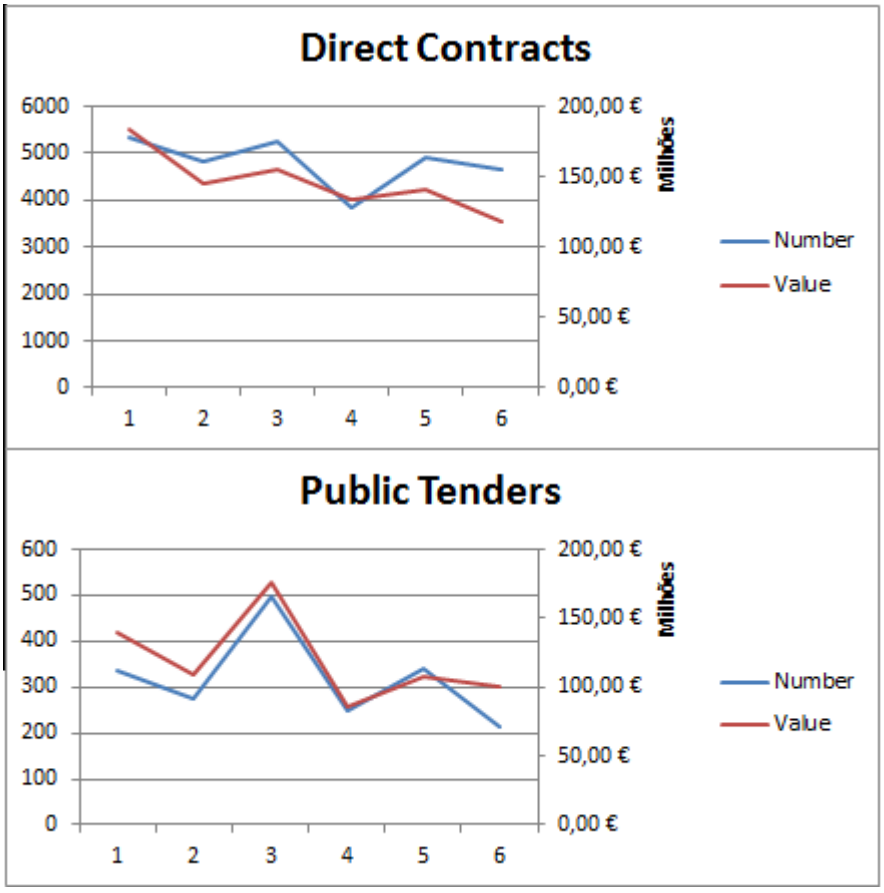


Figure 5.12 – Number and value of direct and public tender contracts in 2012

As an important feature, analysis per business area is available. The user can select a business area of interest and related information is displayed:

Table 5.8 - Public contracts' data according a specific CPV category

Select CPV main category	45	Construção

2012 General Data

		2012						
		Total	01	02	03	04	05	06
All Contracts	#	4601	766	729	907	629	754	816
	Total Value	2.536.585.816,36 €	137.297.118,10 €	1.193.586.766,57 €	816.040.654,11 €	87.978.135,21 €	130.145.794,17 €	171.537.348,20 €
	Average Value	551.311,85 €	179.239,06 €	1.637.293,23 €	899.714,06 €	139.869,85 €	172.607,15 €	210.217,34 €

Main clients within the business area are presented according to the number of contracts or the total value:

Table 5.9 - Top clients according to the total contract value

Position	NIF	Client	Contracts	Total Value
1	508069645	PARQUE ESCOLAR EPE	200	1.784.074.221,16 €
2	503933813	REDE FERROVIARIA NACIONAL REFER EP	135	154.904.483,23 €
3	507866673	REN REDE ELECTRICA NACIONAL SA	45	93.018.972,24 €
4	500051070	MUNICIPIO DE LISBOA	464	34.989.837,13 €
5	503876321	A 505600005 SULDOURO VALORIZACAO E TRATAMENTO DE RESIDU	33	25.627.497,12 €
6	506656128	MUNICIPIO DE PAREDES	157	23.429.916,08 €
7	500700834	ANA AEROPORTOS DE PORTUGAL SA	86	22.382.236,41 €
8	503148776	ADMINISTRACAO REGIONAL DE SAUDE DE LISBOA E VALE DO TEJO II	602	21.684.593,97 €
9	512091773	ATLANTICOLINE SA	15	19.477.743,24 €
10	504598686	EP ESTRADAS DE PORTUGAL SA	113	17.188.311,72 €
11	501449752	APDL ADMINISTRACAO DOS PORTOS DO DOURO E LEIXOES SA	107	16.813.932,81 €
12	500792771	BANCO DE PORTUGAL	176	16.524.881,45 €
13	506110508	AGUAS DO AVE SA	7	15.687.650,45 €
14	508053307	EMA EMPRESA DE MEIOS AEREOS SA	11	14.933.943,50 €
15	501294163	MUNICIPIO DE ANADIA	63	14.796.749,36 €
16	501158740	MUNICIPIO DE ESPINHO	23	13.968.221,93 €
17	501073663	MUNICIPIO DE PENAFIEL	114	13.788.674,69 €
18	506901173	MUNICIPIO DE BRAGA	77	12.595.356,79 €
19	672002426	SECRETARIA REGIONAL DO AMBIENTE E DO MAR	38	12.574.382,38 €
20	501356126	INSTITUTO NACIONAL DE EMERGENCIA MEDICA IP	249	12.508.739,10 €

The presented graphics and table serve to demonstrate possible interpretations of data. More valuable configurations of data organization, visualization can be proposed to enhance decision making support value.

5.2.3 Discussion

Objective achievements

The software achieved a very good efficiency and effectiveness in data collection, mostly because the crawler and wrapper processes are very simple. The crawler uses a unique URL from which requests search results for a given date. The output of the request is an already structure information file. The number of invalid instances is negligible (9 in 30930) and resulted because of the inexistence of supplier. Other instance unexpected values such us a null contract price. This may

result from a lack of information from the target web page or the organization responsible to send the information to base.gov.pt.

Although is intended to sharpen the NIF extraction to clear format errors the software configuration ensures a good level of integrity.

Multiple configuration possibilities

A very interesting multiple configuration is possible combining the first case study with the present one. Using ontology capabilities, it would be possible to track the public tender results with the public tenders' proposals. Then it would be possible to study interesting KPI's. For instance, the gap between the base values of a public tender with the final contract price would be a very interesting study matter.

For this configuration to be possible a third software element would have to be program or integrated in one of the software's. The software would gather information from the ontologies and reason about the content. To track public contract to the original tender proposal additional information would have to be collected from the second software. This additional feature should be integrated in the already created software.

Market value potential

The application's value is similar to the first case study. The application is capable of continuously and autonomously collect web data and process it into intelligence.

In this case, the subject is public contract. The insights harvested with PC information represent the real transaction streams that sprout from all public organizations. Organizations using this tool are able to fully characterize public spending in different business markets. Intelligence regarding the organization's business area is important to characterize the market environment, competitors' behavior and analyze public spending has a whole.

Insights from data analysis are able to determine if the best strategy is in practice and support new configurations for changes in public spending course. Changes in strategic position can be tracked to evaluate performance and support future direction. This demonstrates adaptation and, therefore, dynamic capabilities are improved supporting the organization's survival ability.

Chapter VI – Conclusion and future work

6.1 Conclusions

The development of the software prototypes showed it is possible two create simple, efficient and reliable application responsible for gathering, filtering and store web information for data analysis with competitive intelligence value following the proposed methodology. Those tasks incorporate all necessary steps to develop a solid application capable of running in a simple machine and collected web data. It is very important to propose a solid and simple competitive intelligence objective in order to develop a high efficiency and robust application.

Both cases demonstrated a high level of effectiveness with minimal errors, most associated with problems of information structuring from the web site side. Errors can be tracked when all documents are saved in memory. It is proposed to create a number of simple data format checking to ensure the desired information was correctly obtained. When dealing with organization's names is suggested to find an alternative identifier. Names are very liable to have equivalent strings. In our case studies, we have collected the financial identifier of every client/supplier which allowed us to ambiguously filter information.

Although no user interface was developed due to time constrains, data analysis are viable and reflect the main value component if they are capable of transforming information into intelligence. It is also of great importance that information treatment effectively produces intelligence value. Those analyses may be proposed by management which is interested in implementing one application.

Another interesting conclusion in this dissertation was the high level of resemblance between the two software's' architecture. The development of the second case study showed a high level of similarity between software architecture suggesting not only that the methodology is solid and replicable but also that a common software structure can be proposed. For instance, crawler and wrapper are highly integrated; common phases of information treatment and document handling are present. The applications have three clearly distinct elements as suggested in the initial framework (figure 6): data gathering, storage of information, and data presentation.

Regarding crawler developments, we strongly support the one project model instead of using open source crawlers. Open source crawlers have to be prepared to deal with a great amount of internet interactions. Studying a specific web target one simple method may be capable of accessing and searching for target documents. In this case, crawler algorithms are simple to develop and

implement. When enough applications have been developed, a library of crawler algorithms can be created and used for future developments.

Regarding the value of the two applications we think we demonstrated two value components. The first is the competitive advantage that one organization can withdraw from data analysis and how it can support strategic decision making. The second has functional value in internal procedures if the application itself has capabilities already developed in the organization process. Information search by human work can be automatically processed by the application and workforce can focus on other tasks.

The advantage withdrawn from gathered intelligence should help a company to better align their policy to the market environment. The software enables a great amount of information to be automatically gathered with minimum human intervention. This information is filtered, treated and visually organized in order to characterize market behavior, competitors power and evaluated the organization's performance. Those capabilities should support top level decision making.

The market potential of the functional capabilities is justifiable with a more efficient internal procedure, decreasing time wasting and diverting some workforce to more valuable tasks.

The application helps achieving superior competitive advantage if it is capable of support strategic decision that enhances survival capabilities. All four information gaps suggested by Rohrbeck (2010) are somehow filled with the development of such applications. They enhance *Reach* capability since they can be responsible of gathering information from otherwise difficult to intensively search web sites. *Scope* if they intend to widen the business related information. *Time Horizon* element is present by the storage of all gathered data allowing tendencies to by study. Multiple configurations of software modules can decrease the *Source* gap. Humanly impossible information search can now be easily achieved with the right deployment and with pertinent intelligence objectives.

If strategic direction changes according to collected intelligence, the competitive results can be observed in future intelligence and help evaluate those decisions. Implementing one application not only helps characterized a given market and competitors but also evaluate is the correct strategic decisions are in place. This constant evaluation of strategic decisions and subsequent adaptations to market and competitors behavior is a dynamic process. Hence, we can considerer that dynamic capabilities are improved by the methodology.

For all the presented work we considered of great value the presented methodology.

6.2 Recommendations for future work

Since the dissertation was developed in a business environment with no IT experts support there is no assurance that state-of-art programming practices were used. We suggest the development of application in other computer languages such as Java that represent a higher level of compatibility with different operating systems. Tests with a working application running on internal servers or as web based software are suggested.

From the similarity between the two applications it is also interesting to study if it is possible to create a ground base architecture. For instance, an application from which variables could control all its behavior. The application could have inputs such as the ontology scheme file, URL, second level URLs, directories for files storage, converting procedures, and finally, ontology storage. The same application could be used to create any Web Competitive Intelligence tool.

The developed applications were intended to be located in local machine with internet access. It would be of great interest to transform the applications into web services from which companies could ask for competitive intelligence services. A *Service-as-a-Software* could be developed. In the front end, customers would request specific competitive intelligence tasks and received results. In the back end, using WeCIM, local applications would collect information according to requests and launch treated data to the server to be accessed by the client. For example, a company could request all prices of all scanners in a given web site. In this case, the base architecture would accelerate the applications development and deployment.

Regarding crawlers development, we suggest that a library of crawling algorithms can be created to assist future applications' development. When a new application is in studying phase, an old algorithm may suit the solution by changing some inputs.

Other interesting matters are the use and reuse of already formulated ontologies to accelerate the process of software development and enrich capabilities. The global acceptance of the Semantic Web practices could also improve the development of such applications since web sites would be ready to be semantically searched and information would be retrieved much easily.

Ontologies can also play a different role within applications. Linguistic ontologies could correlate specific words relations, such as being equivalent or opposite in meaning, and helping developing application with more semantic capabilities. In our case studies simple structured information types were collected but if it was intended to collect comments posted by users, the application had to have NLP capabilities to determine if users denote positive or negative thoughts.

It is also of great importance the development of a high level ontology that could represent the enterprise knowledge field. That ontology should be integrated in a base framework upon small application modules could be implemented. This complete approach sustains not only competitive intelligence objectives in different sectors of organizations but also internal processes according to business models and strategic premises.

It is intended to proceed with the deployment of one of the case studies in real environment to prove competitive advantage capabilities.

Bibliography

Ansoff, H. Igor. “Strategic issue management, Strategic issue management.”, *Strategic Management Journal* 1, vol. 2, pp. 131–148, April 1, 1980.

Ashton, W.B, and G.S. Stacey, “Technical Intelligence in Business: Understanding Technology Threats and Opportunities”, *International Journal of Technological Management* 10, vol. 1, pp. 79–104, 1905.

Batavick, F., and H.C. Lucas, *The Transformation Age: Surving A Technological Revolution with Robert X*, Maryland Public Television and University of Maryland’s Robert H Smith School of Business: Cringely (DVD), 2008.

Baumgartner, Robert, Oliver Frölich, Georg Gottlob, Patrick Harz, Marcus Herzog, Peter Lehmann, and Tu Wien, “Web Data Extraction for Business Intelligence: The Lixto Approach.”, In *In Proc. of BTW 2005*, pp. 48–65, 2005.

Becker, Patrick. “Corporate Foresight in Europe : A First Overview.” *Science and Technology*, pp. 27, October 2002.

Benjamin, I. Robert, *Information Technology: a Strategic Opportunity*, Center for Information Systems Research, Massachusetts Institute of Technology, Sloan School of Management, 1983.

Berners-Lee, Tim, James Hendler, and Ora Lassila, “The Semantic Web”, *Scientific American*, May 2001.

Cepeda, Gabriel, and Dusya Vera, “Dynamic Capabilities and Operational Capabilities: A Knowledge Management Perspective”, *Journal of Business Research* 60, vol. 5, pp. 426–437, 2005.

Christensen, Clayton M, *The Innovator’s Dilemma: When New Technologies Cause Great Firms to Fail*, 1st ed., Harvard Business Review Press, 1997.

Day, George S., and Paul J. H. Schoemaker, *Peripheral Vision: Detecting the Weak Signals That Will Make or Break Your Company*, 1st ed. Harvard Business Review Press, 2006.

Deutsch, Claudia H. “At Kodak, Some Old Things Are New Again”, *The New York Times*, section Technology, May 2 2008 http://www.nytimes.com/2008/05/02/technology/02kodak.html.

Eisenhardt, Kathleen M, and Jeffrey A Martin, “Dynamic Capabilities: What Are They?”, *Strategic Management Journal* 21, vol.1, pp. 1105–1121, October 2002.

Fine, Charles H. *Clockspeed: Winning Industry Control in the Age of Temporary Advantage*, 1st ed., Basic Books, 1998.

Fuld, Leonard M, *The New Competitor Intelligence: The Complete Resource for Finding, Analyzing, and Using Information About Your Competitors*, 2nd ed., Wiley, 1994.

Gómez-Pérez, Asunción, Mariano Fernández-López, and Oscar Corcho, *Ontological Engineering*, 2nd ed., Springer-Verlag, 2004.

Griffith, David A, and Michael G Harvey, “A Resource Perspective of Global Dynamic Capabilities”, *Journal of International Business Studies* 32, vol. 3, pp. 597–606, 2001.

Guarino, Nicola, and Pierdaniele Giaretta, “Ontologies and Knowledge Bases: Towards a Terminological Clarification”, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pp. 22–35, 1995.

Jain, Subhash C, “Environmental Scanning in U.S. Corporations”, *Long Range Planning* 17, vol. 2, pp. 117–128, April 1984

Kantrow, A M, "The Strategy-Technology Connection", *Harvard Business Review* 58, vol. 4, pp. 6–21, 1980.

Kessler, Eric H., and Alok K. Chakrabarti, "Innovation Speed: A Conceptual Model Of Context, Antecedents, And Outcome", *Academy of Management Review* 21, vol. 4 pp. 1143–1191, October 1996.

Lee, H., K. G. Smith, and C. M. Grimm, "The Effect of New Product Radically and Scope on the Extent and Speed of Innovation Diffusion", *Journal of Management* 29, vol. 5, pp. 753–768, October 2003

Lucas Jr., Henry C., and Jie Mein Goh, "Disruptive Technology: How Kodak Missed the Digital Photography Revolution", *The Journal of Strategic Information Systems* 18, vol. 1, pp. 46–55, March 2009

Miller, Jerry P., *Millennium Intelligence: Understanding and Conducting Competitive Intelligence in the Digital Age*, 1st ed., New Jersey: CyberAge Books

Pina e Cunha, Miguel, and Chia, Robert, "Using Teams to Avoid Peripheral Blindness", *Long Range Planning* 40, vol. 6, pp. 559–573, December 2007

Porter, Michael E., "How Competitive Forces Shape Strategy", *Harvard Business Review* 57, vol. 2, pp. 137–145, 1979.

Porter, Michael .E., and Victor E. Miller, "How Information Gives You Competitive Advantage", *Havard Business Review* 63, vol. 4, pp. 149–160, 1985.

Reger, G., "Strategic Management of Technology in a Global Perspective: Differences Between European, Japanese and US Companies", *PICMET - Portland State Univ*, pp. 20, 2001.

Reimer, Ulrich, Peter Brockhausen, Thorsten Lau, and Jacqueline R. Reich, "Ontology-based Knowledge Management at Work: The Swiss Life Case Studies", In *Towards The Semantic Web: Ontology-Driven Knowledge Management*, 1st ed., Wiley, 2003.

Rohrbeck, René. *Corporate Foresight: Towards a Maturity Model for the Future Orientation of a Firm*, 1st ed., Springer, 2010.

Rouach, Daniel, and Patrice Santi, "Competitive Intelligence Adds Value: Five Intelligence Attitudes", *European Management Journal* 19, vol. 5, pp. 552–559, October 2001

Shapiro, Carl, "The Theory of Business Strategy", *RAND Journal of Economics* 20, vol. 1, pp. 125–137, 1989.

Sood, Ashish, and Gerard J. Tellis, "Technological Evolution and Radical Innovation", *Journal of Marketing* 69, pp. 152-168, 2005.

Swasy, Alecia, *Changing Focus: Kodak and the Battle to Save a Great American Company*. 1st ed., Crown Business, 1997.

Teece, David J, Gary Pisano, and Amy Shuen, "Dynamic Capabilities and Strategic Management", *Strategic Management Journal* 18, vol. 7. pp. 509–533, August 1997.

Uschold, Mike, and Martin King, "Towards a Methodology for Building Ontologies", In *Workshop on Basic Ontological Issues in Knowledge Sharin, Held in Conduction with IJCAI-95*. Montreal, Canada, 1995.

Winter, Sidney G., "Specialized Perception, Selection, and Strategic Surprise: Learning from the Moths and Bees", *Long Range Planning* 37, vol. 2, pp. 163–169, April 2004.

Zollo, Maurizio, and Sidney G Winter, "Deliberate Learning and the Evolution of Dynamic Capabilities", *Organization Science* 13, vol. 3, pp. 339–351, May 2002.