

developing business intelligence solutions using the only 100% open source BI suite

BUSINESS INTELLIGENCE WITH SPAGOBİ

SPAGOBİ COMPETENCY CENTER



Open Source Business Intelligence

data mining external processes charts ad hoc reporting reporting
free inquiry real time location intelligence collaboration olap
kpi office etl cockpits mobile master data management

SPAGOBI COMPETENCY CENTER, ENGINEERING

Business Intelligence with SpagoBI

Copyright © 2012 Engineering
All Rights Reserved
SpagoBI Competency Center
Corso Stati Uniti 23/c
35127 Padua, Italy
spagobi@eng.it
www.spagobi.org

Table of Contents

About the authors	VII
Lecture notes	VIII
Introduction.....	X
1. The SpagoBI Project.....	1
Open Source Business Intelligence.....	1
SpagoBI Suite.....	2
SpagoBI history	3
Open Source Strategy	7
Development Strategy.....	8
Ecosystem and Community.....	12
2. SpagoBI Architecture.....	14
SpagoBI Server	16
SpagoBI Meta	17
SpagoBI Studio.....	18
SpagoBI SDK	18
SpagoBI Applications	18
3. SpagoBI Meta	20
Goals and targets.....	21
Installation and configuration	22
The business model.....	32
Creation of the business model.....	34
Inquiring the business model	41

Deploy and use the business model	46
4. SpagoBI Studio	50
Goals and targets.....	51
Installation and configuration	54
Common tasks	61
Data set definition	61
Download and deploy.....	63
Developing documents.....	66
Report	68
Chart	92
Real time.....	122
Cockpit	130
Location Intelligence.....	137
5. SpagoBI Server	152
Goals and targets.....	154
Architecture.....	155
Delivery layer.....	156
Analytical layer	157
Data layer	159
Administrative tools and cross services.....	159
Multi-tenancy.....	161
Installation and configuration	162
The behavioral model	168
User and role configuration	169
Repository structure and rights	175
Analytical drivers	178
Publication environment	191
Analytical model.....	196
Main concepts	196
Document types.....	198
Register an analytical document.....	200
Correlation between analytical drivers	210

Controlled Visibility	212
Cross services.....	213
Document browser	213
Subscriptions	227
Massive export and scheduling	229
Administrative Tools.....	232
Scheduler.....	232
Import/export.....	246
Engine management.....	249
Map catalogue	255
Data sources	256
Data sets.....	260
6. Analytical engines.....	277
Report.....	278
SpagoBIBirtReportEngine.....	279
SpagoBIJasperReportEngine.....	280
SpagoBIAccessibleReportEngine	281
Multidimensional analysis	285
SpagoBIJPivotEngine	286
SpagoBIJPaloEngine.....	310
SpagoBIJPXMLAEngine	318
Chart	322
SpagoBIJFreeChartEngine.....	323
SpagoBIJSChartEngine	363
Analytical cockpit.....	399
SpagoBICompositeDocEngine.....	400
KPI.....	403
Build a KPI model.....	405
Calculate a KPI model.....	426
Create a KPI analytical document	427
View a KPI model.....	432
Data Mining	433

Basic concepts	434
SpagoBIWekaEngine	436
Advanced functionalities	448
Free Inquiry	455
SpagoBIQbeEngine	456
SpagoBISmartFilterEngine	480
Ad hoc reporting.....	491
SpagoBIWorksheetEngine.....	492
Location Intelligence.....	501
Basic concepts	502
SpagoBIGeoEngine	507
SpagoBIGEOReportEngine.....	533
Real time	551
SpagoBIDashboardEngine	552
SpagoBIConsoleEngine.....	563
Mobile BI	620
Architecture	620
Installation	622
Navigator.....	623
SpagoBITableMobileEngine	625
SpagoBIChartMobileEngine	629
SpagoBICockpitMobileEngine	635
Office integration.....	639
SpagoBIOfficeEngine.....	639
Collaboration	641
SpagoBIDossierEngine	642
7. Operational engines	652
ETL.....	652
SpagoBITalendEngine.....	652
External processes	662
SpagoBIProcessEngine	662
Master data management.....	670

SpagoBIRegistryEngine.....	670
8. SpagoBI SDK.....	676
Goals and targets.....	676
Installation and configuration	677
SDK Web Services	682
Proxy.....	682
DocumentService	683
DataSetsSDKService.....	699
DataSourcesSDKService.....	704
EngineService.....	705
MapsSDKService.....	707
SDK Tag Library.....	709
SDK Javascript APIs.....	712
9. SpagoBI Applications	715
Goals and targets.....	715
Installation and configuration	717
Audit and monitoring	719
Audit documents	720
The last access point for the analysis is the document showing the list of LOVs. LOVs can be searched using a filter on the type of LOV, namely: query, fixed value, and script.....	725
Monitoring documents	727
Other SpagoBI Applications	727
Spago4Q.....	727
SpagoBI for Areas	728
SpagoBI for Elisa.....	729
SpagoBI for Adaxa Suite	730
Reference and resources.....	732
General information.....	732
Technical resources	733
Professional services.....	733
Figures and Tables	735

Index of figures	735
Index of tables	743

About the authors



Grazia Cazzin

Grazia, SpagoBI Project Leader and SpagoBI competency center director, is a technical manager at Engineering Group. With many years of experience in enterprise application development, data modeling, data warehousing, dimensional analysis and business intelligence, she has gained valuable expertise working in several market sectors (industry, finance, public administration), covering several thematic areas (ERP, MRP, MPS, Enterprise Portals, CRM, DWH and BI). She has served as Assistant Professor of Business Intelligence and DSS at the Faculty of Mathematical, Physical and Natural Sciences of the University of Turin (Italy).



SpagoBI Core Team

SpagoBI core team includes consultants, architects and developers that constantly work on SpagoBI code or projects, mixing technical competencies and business intelligence knowledge. Team members have significant experience in several market sectors, e.g., industry, finance, public administration, and healthcare. They contribute to SpagoBI growth by sharing their competencies and by proposing novel ideas and emerging topics in BI.

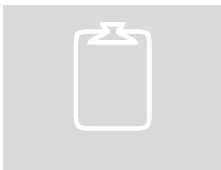
Lecture notes

Some graphical conventions have been adopted to allow readers to easily identify special contents such as notes, summaries, essential information and so on. All conventions are explained hereafter.



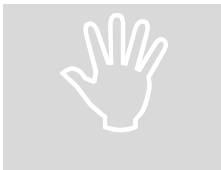
Notable content

This icon highlights relevant content.



Summary

This icon highlights short texts that sum up key messages.



Attention

This icon warns the reader about possible errors and problems using SpagoBI.



Advice

This icon identifies suggestions to better use SpagoBI.



Writing convention

This icon explains any stylistic choice.



Read more

This icon refers to additional documentation that may be useful to the reader. It usually points to external sources for non-native SpagoBI features.



Reference

This icon points to another section of the book dealing with a specific subject.



Work in progress

This icon highlights SpagoBI functionalities that are still under development or need improvement.

The following fonts have been conventionally adopted, to easily identify special words and expressions:

- **Menu**, **menu item** and **static label** directly refer to SpagoBI GUI
- **Input field** is a label referencing input fields in SpagoBI GUI
- *SpagoBI keyword* highlights SpagoBI terminology
- `Code example` is a piece of code showing configuration patterns or parts of a document template.

Introduction

Business Intelligence platforms enable users to build information support applications that help organizations in their decision-making processes for enterprise strategic and tactical management.

The BI market has recently witnessed two relevant trends. In the proprietary world, major IT vendors have been acquiring existing BI products, thus creating a small set of leading actors sharing the market. On the other hand, open source BI systems have emerged as a suitable solution for companies wishing to combine specific features from different tools, instead of adopting monolithic proprietary BI products.

Today SpagoBI is the only 100% open source, complete and flexible business intelligence suite. SpagoBI covers all areas of Business Intelligence with a wide set of analytical engines and cross functionalities. The suite has been developed to constantly meet real end-users' needs, carefully balancing open source flexibility and enterprise-grade software quality. Until now SpagoBI has been used in dozens of BI projects, making a constant effort to find the best solution for the end-user, saving time and economic resources.

After seven years spent managing BI projects with SpagoBI and developing the suite, time has come to share the acquired knowledge and expertise, letting more and more users get to learn SpagoBI and make the best use of its various features.

This book is a reference manual for those wishing to learn how to use SpagoBI. It targets different types of users, just as the different modules of SpagoBI do. All modules are described from the perspective of the main target user, showing examples and providing useful hints and references. Readers can choose whether to take a quick overview on the suite, or follow the detailed explanation of a specific tool.

Needless to say, a book cannot replace those design and analysis activities that are fundamental to the success of any BI project. Nevertheless, this manual represents a valid support for BI developers, integrators, domain experts and users interacting with SpagoBI in any phase of the project.

The book is organized as follows:

Chapter 1 provides an overview of SpagoBI project, its history and specific characteristics, its open source and development strategies, the ecosystem and SpagoBI community.

Chapter 2 briefly introduces SpagoBI suite architecture and its composing modules, which are described in the following chapters.

Chapter 3 presents SpagoBI Meta, the graphical environment to build and inquire business models.

Chapter 4 describes the environment for developing SpagoBI analytical documents: SpagoBI Studio. It provides installation and configuration instructions, as well as detailed explanations for each document designer.

Chapter 5 and 6 are focused on SpagoBI Server. The first presents its overall architecture, cross services and administrative tools, as well as the main conceptual models driving its logics. This includes the behavioural model, which rules visibility and permissions on documents; and the analytical model, a common framework to represent the BI project.

All analytical areas covered by SpagoBI are described in Chapter 6, which focuses on the main functionalities provided by each engine and offers a detailed step-by-step guide to build documents.

Chapter 7 is dedicated to users wishing to integrate SpagoBI with their own information system by relying on SpagoBI SDK.

Chapter 8 presents SpagoBI Applications through relevant examples.

Finally, useful references and resources are provided.

1. The SpagoBI Project

Open Source Business Intelligence

Business Intelligence platforms enable users to build information support applications that help organizations in their decision-making processes for enterprise strategic and tactical management. It is a vibrant and constantly growing sector in the Information Technology market, as confirmed every year by the main IT research and advisory companies. The strong competition that characterizes this market led to several acquisitions over time, until the outline of a few mega-vendors, which control almost all proprietary offering.

During the last years, the landscape of business intelligence platforms and products has undergone a radical process of transformation. Some historical suppliers merged creating a narrow circle of leading actors, while new models emerged aiming at reducing the total cost of acquisition and management as a whole: software as a service, cloud, mobile applications, appliances and analytics.

In this scenario open source plays a key role as it allows users and organisations to combine the best features of available tools, functionally enriching and adapting them to the typical architectural contexts of business intelligence enterprise solutions, which are strictly related to issues such as security, scalability and reliability.

In the past, good open source tools were provided that focused on specific Business Intelligence tasks, such as:

- JasperReport and BIRT as reporting tools
- JFreeChart as chart library
- Weka as data mining tool
- Jpivot/Mondrian as OLAP tool
- Kettle and Talend as ETL tools.

Each single tool was satisfactory in its domain, but its usage was limited to personal use. In fact all solutions were developer-oriented and they did not cover enterprise-level issues such as security, life cycle, scalability and delivery.

Starting from those tools, some Business Intelligence products appeared on the market that integrated them into an enterprise architecture. Each Business Intelligence product enhanced its approach and features over time, providing new and different BI capabilities released with different policies, according to the vendors' different business models.

Today, SpagoBI is the only entirely open source business intelligence suite adopting a peculiar approach that makes it unique in this domain.

SpagoBI Suite

SpagoBI is the only 100% open source, complete and flexible business intelligence suite.

SpagoBI is 100% open source forever, because it is released under an OSI¹ approved open source license, protecting the freedom of the code and allowing its commercial use, resulting in the development of various business models in its own eco-system. It adopts the “pure open source” business model as it is released as only one stable version. It is an alternative to the *dual-licensing* or *open core* model, which also includes a parallel proprietary software release as an enterprise or professional version. In order to guarantee the open source transparency, openness, quality, sustainability and availability over time,

¹ <http://www.opensource.org/>

SpagoBI software is download-able from the forge of OW2 Consortium², a no-profit and independent open source community.

SpagoBI is a complete suite because it covers the whole range of Business Intelligence needs, with innovative solutions, providing users with a wide range of analytical tools supporting developers, testers and administrators in their daily work. This maximizes the effectiveness of data analysis according to the principle: “the right tools for the right user”.

SpagoBI is a flexible suite, because it offers many engines for the same analytical area, allowing users to freely choose how to build their own solution. Based on open standards, avoiding a pre-defined software stack, SpagoBI provides the best solution for users’ needs, saving investments and providing quick results. The number of available engines and their characteristics have been growing over time according to SpagoBI project road-map. Moreover, the suite is suitable to various environments, as it can easily integrate with security systems, portals and legacy applications.

The suite is complemented and supported by a complete set of professional services, including training, consulting, development, support and maintenance, provided by SpagoBI Competency Center.

SpagoBI history

SpagoBI belongs to SpagoWorld³, the open source initiative by Engineering Group. The story began in 2001, with the need of the company to realize a Java Enterprise framework that could support project development for its customers. At the end of 2004, this framework was re-engineered and released as open source software on SourceForge - the Spago framework. SpagoBI

² <http://www.ow2.org>

³ <http://www.spagoworld.org>

project⁴ started in late 2004. In July 2005 SpagoBI was released for the first time on SourceForge, and then moved onto the forge of ObjectWeb Consortium. In 2006 SpagoWorld initiative was formally launched and the codebase of SpagoWorld projects was moved onto the forge of the new OW2 Consortium. In 2007 two more projects were released: Spagic, the open source universal middleware, and Spago4Q, which is a vertical application of SpagoBI focused on quality of products, processes and services. In 2010, when Engineering Group joined the Eclipse Foundation, the eclipse eBPM and eBAM projects, the latter lead by SpagoBI team, and the GeoBI initiative, were launched.



FIGURE 1.1 – SpagoBI history

Nowadays SpagoBI Competency Center is a work unit of the Research and Innovation division of Engineering Group managing all the activities related with SpagoBI suite and open source business intelligence: development and management of SpagoBI and eBAM projects, supply of SpagoBI professional

⁴ The name SpagoBI refers to SpagoWorld initiative, marked by the root of the word "Spago". Spago (i.e., twine) is not an acronym, but an Italian name suitable for an integration approach (it ties, without compelling). It may also recall the widely known, fine Italian cuisine: well, if you like Spaghetti, Spago software may be an excellent choice for you!

services, management of GeoBI initiative, participation in BI-related OW2 Consortium initiatives, development of business intelligence projects and BI consultancies at Engineering Group.

When the first version of the Open Source Business Intelligence SpagoBI suite was released in 2005, within the framework of SpagoWorld initiative, the aim of the SpagoBI project was to build an integration platform which could offer a flexible and sustainable BI solution to organizations and companies, supporting the measurement and management of their critical assets and success factors.

At the beginning, the product was built by integrating the best available open source tools, making them usable at the enterprise level, by customizing and extending their basic functions on the basis of the information coming from the first business intelligence projects in which the product itself was used. This way, SpagoBI development team created new analytical engines and consequently started a virtuous circle, according to which BI project needs quickly turned into additional practical capabilities or features that contributed to the improvement of the product.

As a result, the product has become more and more powerful and usable. Specifically, innovative practical functionalities have been added at a first stage, which was substantially oriented towards the support of the basic business intelligence functions, such as reporting and multidimensional analysis. They included: representation of data analysis on geo referenced supports (Location Intelligence), data inquiry and collaborative functions, using annotations and cooperative workflow.

The first wave of projects realized by means of SpagoBI version 1.x was mainly motivated by users' wish to test this open source product with a very cautious but also innovative approach. Therefore, the first projects aimed at demonstrating that the product was able to cover BI users' real needs at a much more reasonable cost than proprietary solutions. As time goes by and with the increasing diffusion of the open source model and applications, users' and companies' approach to open source products has started to change. This applied to SpagoBI as well: beside testing strategies, the replacement of proprietary tools and the strategic adoption of open source platforms have become key decision drivers for SpagoBI adopters.

This was just the right context for a significant evolution of SpagoBI towards 2.x version, released at the end of January 2009. This made a considerable step ahead not only from the technological and architectural perspective, but also from the functional one.

From the architectural point of view, three important changes deeply affected the design of SpagoBI core:

- the components splitting towards a fully compliant Oriented Architecture: all features of the previous version were collected in SpagoBI Server and related to each other in a secure mode; in addition, external services were provided in the new SpagoBI SDK module;
- the integration of the CAS security module, which considerably improved access security in all suite services;
- the independence from external systems, such as portals and profiling systems, as well as CMS as the internal objects repository.

From a functional viewpoint, various BI capabilities have been added, such as the new GIS engine for the representation of the analysis on geographical maps; composed documents, allowing to build highly efficient and interactive dashboards; the KPI engine; new dashboards; subscription functions. Next to the Server and SDK modules, the SpagoBI Studio and Applications modules were also first introduced in the 2.x series, as the development environment and a set of pre-built analytical models, respectively.

It is important to notice that, once released, SpagoBI 2.0 version has been used in several business intelligence projects, directly managed by SpagoBI development team, ensuring solidity and response to the requirements of Engineering Group's major customers.

Finally, in mid-June 2011, version 3.0 of SpagoBI was released, marking an important turning point in the domain of enterprise-grade open source solutions. This version offers a module fully dedicated to metadata management (SpagoBI Meta), allowing users to define different information models within their own data domain and to develop new analytical documents through the complete data sources abstraction. It also includes a new ad-hoc reporting engine for the guided creation of customized reports and a new live-chart engine supporting the realization of advanced cockpits even on mobile

devices, such as iPad and Android tablets, as well as iPhone and Android smart phones. The 3.0 release also improved real-time data analysis, based on systems and events interacting with OLTP (On-line Transaction Processing) applications. The real time console was integrated in the framework of a new Eclipse project named extended Business Activity Monitoring (eBAM), where a Complex Event Processing (CEP) engine was also implemented.

With the 3.x series, SpagoBI has not only grown to become a complete BI suite, but it has also increased the range of solutions and services to support users in the adoption of the product (agile development methodology, on-demand, cloud and SaaS usage models), while enriching its offer of support and training services.

Today SpagoBI is an open source business intelligence suite providing features comparable with proprietary products, such as Agile BI, Mobile BI, Cloud BI, Real-time BI, and offering vendors and integrators a new window of opportunities in the Information Technology market.

Open Source Strategy

SpagoBI open source strategy is based on the following key principles:

- the development of enterprise-grade software supported by companies and communities (both industrial and open), allowing users – enterprises and organizations – to use mature, stable, scalable solutions and a wide range of support services;
- the full adoption of the open source model in order to foster knowledge and expertise exchange, which is a crucial element in a really free and open source environment;
- the awareness that open source communities have been evolving over time towards a wider *ecosystem* model, where companies of different size and business activity cooperate to improve open source products and bring them to a full enterprise quality level.

According to this strategy, SpagoBI belongs to a new generation of open source projects, characterized by:

- holistic approach and planned development process;
- suitability to complex technological domains aiming to support the achievement of business goals in various markets;
- mature process development;
- involvement of a wide community of users, companies and organizations, open source specialists, consultants and developers;
- guaranteed support services.

To this end, SpagoBI suite is released under an open source software license (MPLv2⁵) aiming at encouraging a wider adoption of the suite, while clearly stating three basic principles:

- it is both open source and free software (i.e., GNU (X)GPL compatible);
- it is a weak copyleft licence based on the protection of the free code of the suite;
- it is a commercial friendly license: the code of the suite can be “linked” to open source and proprietary third-party software⁶.

Development Strategy

SpagoBI competency center fosters the sustainable development of business intelligence projects and applications, in order to allow organizations and enterprises achieve their mission-critical goals, within a trustworthy environment, guaranteed by the software quality, the use of open standards and open source software, and innovation. The strategy adopted by SpagoBI competency center is based on sharing the developments with the community, consolidating the open source solutions at the enterprise level, strengthening

⁵ <http://www.mozilla.org/MPL/2.0/>

⁶ More information at: <http://www.spagoworld.org/xwiki/bin/view/SpagoBI/Licenses>

international partnerships and orienting the project road-map towards industrial initiatives.

In particular, SpagoBI vision is that user's needs are more valuable than the adopted product, in contrast with many solutions imposing products' constraints onto users' requirements. This way, a good solution can actually improve a business intelligence project, if it promotes:

- a project-oriented solution, which, freed from product fees, favors the project start-up, offering a new balance among the elements involved in the development phase - effort, cost, time and quality of results. This deeply differentiates SpagoBI from many other solutions, which are based on proprietary products and characterized by price discrepancies;
- a coherent design, thanks to the adoption of open standards, a modular architecture, the development of innovative components and the integration of the best open source solutions, supporting their integration and re-use into existing environments, thus increasing the value of the solutions that are already in use;
- an evolutionary or agile development process helping users achieve their goals since their first steps: starting small, thinking big, rapidly obtaining first results;
- a sustainable growth, which allows users to benefit of good features and customizable enterprise-level solutions (which are also high-quality, reliable and scalable), adequate support services at a reasonable cost (including installation/configuration, bug-fixing, maintenance, training, warranty), giving them the opportunity to become confident with open source solutions;
- good features that are actually customizable, adequate support services provided at a reasonable cost, giving users the opportunity to become confident with open source solutions;
- a shared business model among vendors and users in a win-win perspective: sharing business goals and priorities may be the best way to obtain the expected results.

According to this vision, SpagoBI has a distinctive and unique feature: its entirely open source approach, which gets over the cultural attitude of keeping

off new rules and ideas. The most adopted business models in the open source world, known as dual-licensing or open-core, follow the traditional paradigm of the proprietary market. This approach, which considers the product as the central element, relies on license sale as the main source of income and imposes software lock-in, tightly binding subscription services to the rights to use the software.

On the contrary, SpagoBI, thanks to its pure open source approach (i.e. a single stable version of the product, entirely distributed under an open source license, adding no “professional” or “enterprise” proprietary versions), imposes no software lock-in, separating any purchase of services from the right to use the software. Moreover, it adopts a project-centric model instead of a product-centric one. This has influenced the architectural choices, its business evolution, its business strategy and its own approach to the development of business intelligence projects.

The goal is not to make money by selling a product, but to provide new opportunities to create a business intelligence solution that meets end-users’ real needs and fits their budget, involving them in the product development. The pure open source approach adopted by SpagoBI allows the suite to grow over time thanks to the experience and innovative ideas of the SpagoBI development team, which interprets users’ requests and project needs.

This approach works because SpagoBI development team understands users’ needs and transforms them into new requirements to be brought into the project roadmap. Along this process, new ideas are validated until they can be included into the open source solution, allowing to create new value for customers and, through the realized product, for all other users as well. This way, different business intelligence projects contribute in terms of requirements, code, test and funding. Moreover, this model fosters sharing of common developments and research tracks with various organizations and communities.

In the meantime, users asking for support are provided with a complete set of professional services including advice, maintenance, training, consultancy, software development, in-house and on-site project development.

Many SpagoBI innovative engines have been developed in the context of actual business intelligence projects (or software projects including analytical capabilities) required by SpagoBI competency center's customers. They represent the core inspiration source for SpagoBI development and growth. Those projects led, among others, to the development of the Geo Engines, the Composite Document, the KPI, the QbE and the ad-hoc reporting engines.

SpagoBI executive team constantly defines the SpagoBI project road-map as a balanced mix of the market analysis and innovative ideas, always involving customers and partners to create synergies between their needs and the project road-map. Thanks to this approach, customers get answers to their specific needs, and see their money invested in the development of their solution, not in software licenses. They obtain solutions that are technologically mainstream and innovative, being guaranteed about the continuous maintenance of the solution itself even in subsequent SpagoBI releases. At the same time, SpagoBI competency center - the software editor – manages to economically sustain the development of the solution and to constantly orient it to solving actual requirements, enforcing its effectiveness. In a few words, it is a true win-win approach.

This is a significant example of the importance of sharing development and creating synergies between multiple projects. Probably no single software project can support the development of a complete functionality on its own, but splitting investments into multiple projects makes the investments themselves more affordable and sustainable.

SpagoBI project is sustained by Engineering Group, an industrial ICT company. It is a different approach compared to many corporate-lead open source projects that are mainly funded by venture capitalist firms. This is part of the wide open source approach adopted by the company: the project itself must be sustainable, balancing industrial funding, revenues coming from the sale of professional services and contributions coming from the development of business intelligence projects, from the partners and the community. The wide adoption of SpagoBI suite in many enterprise-grade BI environments and its stable maturity demonstrate the effectiveness of this approach.

Ecosystem and Community

SpagoBI is an open source project which guarantees openness, not only thanks to the open availability of its results – mainly source code – thanks to its license policy, but also because its design, collaboration, assembling and adoption in the real world live through the community, which is the main driver of SpagoBI innovation and evolution.

SpagoBI community mainly consists of SpagoBI competency center's customers involved in BI projects, integrator partners and users. All of them create an eco-system where the achieved value is the result of complex interactions among the different participants sharing a common decision space. The final value is the mutual enriching relationship among the different agents composing this network, going far beyond the simple sum of single contributions. The larger the community, the greater the amount and quality of relationships connecting its members, the bigger the shared value.

This model, mainly focused on users' needs and real-world capabilities through the development of business intelligence projects, rather than on the product appeal and marketing, makes SpagoBI one of the best solutions and the most complete suite in the Open Source Business Intelligence domain, offering more flexibility than proprietary business intelligence solutions.

SpagoBI participates in OW2 Consortium, a global, independent not-for-profit open source community, open to companies, institutions and individuals, whose goal is the development of open source code infrastructure (middleware and generic applications). OW2 role is central in SpagoBI open source policy because it guarantees transparency, openness, quality, sustainability and availability of the open source code over time. SpagoBI's active commitment in OW2 Consortium is demonstrated by many activities, above all the following:

- leading – from the technological point of view – the OW2 BI Initiative, a joint effort of OW2 and no-OW2 members, set up to foster the growth of a business ecosystem in the Business Intelligence domain;
- participating in the OW2 Open Source Cloudware initiative, a joint effort of partners from Brazil, China, Europe and the U.S.A., all sharing the same goals in the open source cloud domain.

Moreover, SpagoBI competency center participates in the Eclipse Community with a leading role in the eclipse extended Business Activity Monitoring (eBAM) project, which adds real-time BI capabilities to SpagoBI suite.

SpagoBI project team closely works with a wide network of international partners, which includes systems integrators, technology partners and all those partners who want to contribute to the global success of the suite. All SpagoBI partners contribute to the growth of SpagoBI functionalities and are committed to bring the projects that integrate SpagoBI to rapid success.

Last but not least, SpagoBI ecosystem includes research laboratories and institutes, as well as several universities at the international level, thanks to the continuous support that SpagoBI team offers to researchers, professors and students, both in educational and research activities.

2. SpagoBI Architecture

SpagoBI overall architecture is composed of five modules:



SpagoBI Server, the core of the suite that offers all the core and analytical functionalities



SpagoBI Meta, the environment to define and manage meta-data and business models



SpagoBI Studio, the development environment



SpagoBI SDK, the integration layer that allows external tools and applications to interact with SpagoBI Server



SpagoBI Applications, collecting vertical models focused on a particular business domain.

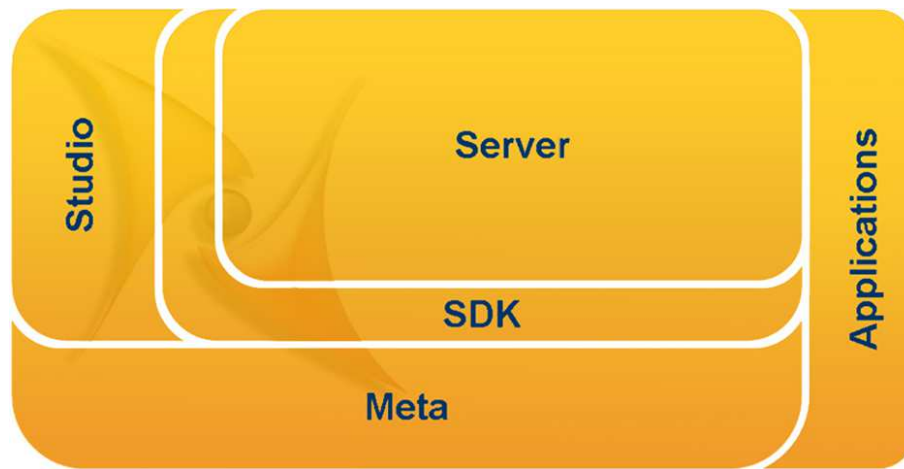


FIGURE 2.1 – SpagoBI suite main modules

The main module, which provides all the BI capabilities, is the SpagoBI Server. This is the only mandatory module, the one that offers a web-based end-user environment to run all types of analysis.

The Meta and Studio modules interact with the Server thanks to the SDK, mainly to authenticate the user, send or receive metamodels, analytical documents or data sets.

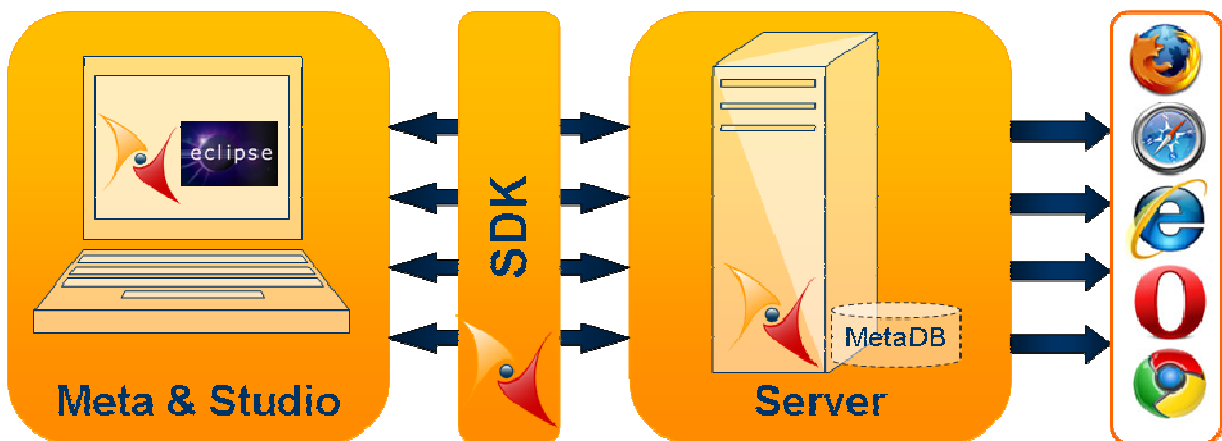


FIGURE 2.2 – Interaction between SpagoBI main modules

A typical BI project uses:

- A single SpagoBI Server : the main BI platform where all BI documents are stored and used by end users. The end-user GUI is entirely web based and can be accessed via a common web browser
- A few instances of SpagoBI Meta: the data manager uses SpagoBI Meta (client application) to define the business models over the enterprise data warehouse
- Several instances of SpagoBI Studio: developers use SpagoBI Studio (client application) to develop analytical documents using the business model, to get data and deploy documents into a remote SpagoBI Server.

The Meta and Studio modules are also available in a single package, which is particularly useful when the data developer is the data manager.

Each module has its own end-user target and covers specific tasks in a development process, as explained in the following paragraphs.

SpagoBI Server

SpagoBI Server is the environment for:

- End-users: they can use their analytical documents by means of this unified access point via a web browser
- System administrators: they can manage the server through a web interface.

For end-users, this module offers all typical BI functionalities :

- Reporting
- Multidimensional analysis (OLAP)
- Charts
- Interactive cockpits
- KPIs
- Data Mining
- Free Inquiry and driven data-selection

- Ad-hoc reporting
- Location Intelligence
- RT dashboards and consoles
- Mobile
- Office automation
- Collaborative tools.

Moreover, some operational BI features are available:

- ETL
- External processes
- Master data management.

For the administrator, SpagoBI Server allows to manage all relevant tasks such as the lifecycle of analytical documents, internal repository, users and roles, security, and presentation environment.

From a technical point of view, SpagoBI Server is a web application deployed into a J2EE application server such as Tomcat, JBoss, WebSphere and WebLogic. It can run on any operating system that supports JVM 1.6 and it works with a private repository hosted on a common RDBMS (MySQL, PostgreSQL, Oracle, Ingres, HSQL, etc.). The application can be accessed through almost all major web browsers (IE, Firefox, Opera, Safari) and it can work as a simple web application or be included in a standard portal server (Liferay, eXo, WebSphere, etc.)⁷.

SpagoBI Meta

SpagoBI Meta is the environment for data owners who define the semantic layer on which the BI documents will be based.

⁷ The complete list of certified environments is available at:
<http://www.spagoworld.org/xwiki/bin/view/SpagoBI/CertifiedEnvironments>

It is a graphical environment to design and inquire business models (semantic layer) and then release them into one or more SpagoBI Servers.

From a technical point of view, SpagoBI Meta is an eclipse plug-in, so it works as a full-client application related to one or more SpagoBI Servers via web services.

SpagoBI Studio

SpagoBI Studio is the environment for BI developers who create analytical documents such as charts, reports and dashboards.

It is a graphical environment to design, test and deploy analytical documents into one or more SpagoBI Servers, using the business model or even directly inquire data sources.

From a technical point of view, SpagoBI Studio is an eclipse plug-in, so it works as a full-client application related to one or more SpagoBI Servers via web services.

SpagoBI Meta and Studio can even be plugged into the same eclipse local installation.

SpagoBI SDK

SpagoBI SDK is the standard development kit for java developers who need to integrate SpagoBI with their applications.

It is a collection of web services, tag libraries or APIs that allow to relate an external application to SpagoBI Server, allowing to use some SpagoBI documents and functionalities in this environment.

SpagoBI Applications

SpagoBI Applications are ready to use as soon as the administrator installs them.

Some pre-built analytical models can be imported by the administrator into SpagoBI Server and be immediately released for the end-user.

An application normally includes:

- Data model (DWH)
- Data or ETL to load data from custom sources
- Analytical documents (reports, charts, cockpits, etc.)

Usually, they don't include code.

3. SpagoBI Meta

In general terms, *metadata* means ‘data about data’ and in the BI domain it usually refers to data used to describe and enrich the original ones in terms of:

- meaning and structure of corporate data
- where they come from
- how they are organized
- how they are valued
- how they are accessed
- how they are used
- how they can be interpreted
- what they are
- what they mean
- what they are for
- etc.

BI literature usually distinguishes between technical and business metadata, according to their nature, use and preferred users.

Technical metadata are commonly used by analysts and developers to build and manage the DWH, and to ensure system security. Therefore, they mainly contain descriptions of data sources (physical patterns), transformations performed on data, constraints and information about data usage.

Technical metadata are used to abstract from data sources, define a business model independently from data storage, generate automatic documentation, build data lineage and manage changes over data structures.

Business metadata are designed to support end users in accessing and using/interpreting business data in the DWH. They typically include many types of re-classification, categorization, description, semantic information, high-level calculation rules.

Business metadata are used to enrich basic data with additional information, categorize documents and provide additional contents to make their interpretation easier.

SpagoBI Meta is the module of SpagoBI suite specifically focused on technical metadata management, whereas business metadata are managed by SpagoBI Server.

Goals and targets

SpagoBI Meta is the module dedicated to technical metadata management and inquiry. It offers a tool supporting the database reverse engineering process, where users can define the semantic layer to be used during the development of analytical documents, so as to define data selection criteria even without any particular expertise on the data domain and query languages.

The target users of the meta module are:

- data owners and data domain experts, who define one or more models that represent informative islands from a business point of view
- administrators who manage or build shared semantic layers under the analytical documents
- developers who inquire semantic layers to build their analytical documents.

In small projects, a single or a few developers are usually in charge of managing all activities. So they can build and inquire semantic layers, playing *de facto* all roles.

Installation and configuration

In order to use the SpagoBI Meta module, the first step is to check that your environment satisfies the following prerequisites:

- it has a JDK 1.6.x already installed
- it has the JAVA_HOME variable already set
- it has a certified operative system (Windows, Linux, Unix are generally accepted)⁸.

If so, the second step is to download the suitable package from OW2 forge at:

- <http://forge.ow2.org/projects/spagobi> to look at SpagoBI latest releases
- http://forge.ow2.org/project/showfiles.php?group_id=204 to choose the latest release among all SpagoBI releases.

Now choose the right package according to the specific release and distribution matching your environment.



SpagoBI Meta version

Using the same version of both SpagoBI Server and SpagoBI Meta is always recommended, even if not mandatory. This is to be sure that both modules are using the same SDK release.

You can choose between:

- SpagoBI Meta, if you are the data owner or the data domain expert and you work on metadata models only

⁸ The list of certified environments mainly depends on the environments supported by Eclipse. Refer to the SpagoBI Meta release notes to know which Eclipse version is included in each released package.

- SpagoBI Studio (which includes the Meta functionalities, too) if you are a developer, the administrator or a one-man-project and you also work on document developments.

SpagoBIPivotEngine-bin-2.0.0_01272009.zip	20,208.3	Any	.zip
SpagoBIObeEngine-bin-2.0.0_01272009.zip	24,318.4	Any	.zip
SpagoBITalendEngine-bin-2.0.0_01272009.zip	4,090.5	Any	.zip
SpagoBIWekaEngine-bin-2.0.0_01272009.zip	10,624.4	Any	.zip
SpagoBI 2.0 - Script db			
mysql-dbscript-2.0.0_01272009.zip	9.7	Any	.zip
SpagoBI 2.0 - Source			
SpagoBI-src-2.0_01272009.zip	242,408.1	Any	Source .zip

c. SpagoBI Meta

SpagoBI 3.3			
SpagoBIMeta 3.3 linux 20111220.zip	367,352.9	Any	.zip
SpagoBIMeta 3.3 linux64 20111220.zip	367,342.9	Any	.zip
SpagoBIMeta 3.3 Win 20111222.zip	370,758.2	Any	.zip
SpagoBI 3.2 RC			
SpagoBIMeta 3.2-RC 11042011 Win.zip	321,758.0	Any	.zip
SpagoBI 3.1			
SpagoBIMeta 3.1 Linux 20110721.zip	315,178.6	Any	.zip
SpagoBIMeta 3.1 Linux64 20110824.zip	315,379.5	Any	.zip
SpagoBIMeta 3.1 Win 20110721.zip	317,571.5	Any	.zip
SpagoBI 3.0			
SpagoBIMeta 3.0 linux 20110620.zip	311,627.4	Any	.zip
SpagoBIMeta 3.0 win 20110516.zip	323,209.8	Any	.zip

d. SpagoBI Studio

SpagoBI 3.3			
SpagoBIStudio 3.3 Linux 20111222.zip	408,092.8	Any	.zip
SpagoBIStudio 3.3 Linux64 20111222.zip	408,073.2	Any	.zip
SpagoBIStudio 3.3 Win 20120120.zip	411,728.7	Any	.zip
SpagoBI 3.2 RC			
SpagoBIStudio 3.2-RC 11042011 Win.zip	361,448.1	Any	.zip
SpagoBI 3.1			
SpagoBIStudio 3.1 Linux 20110721.zip	354,305.7	Any	.zip
SpagoBIStudio 3.1 Linux64 20110824.zip	354,531.3	Any	.zip
SpagoBIStudio 3.1 Win 20110721.zip	355,639.7	Any	.zip

FIGURE 3.1 – Download SpagoBI Meta package from OW2 forge



Eclipse

Both SpagoBI Meta and Studio have been developed as Eclipse plug-ins. Therefore, any limit on supported environments and some technical features and configuration settings derive from Eclipse development model. Please refer to the foundation website at <http://www.eclipse.org/>.

At this point, you just have to unzip the downloaded package under a folder, having a SpagoBIMeta_<version>_<distribution>_<date> subfolder. Here you can find and run a SpagoBIMeta.exe or SpagoBI.sh script to start and select the preferred workspace.

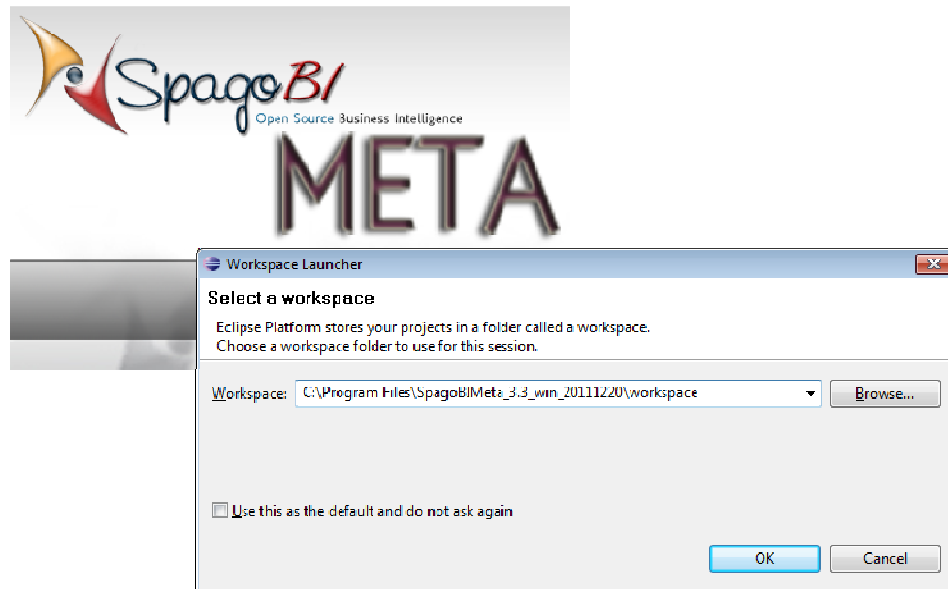


FIGURE 3.2 – Run SpagoBI Meta by selecting the workspace

The selected workspace works as a local repository for all the developed objects (metamodels and data sets). However documents are neither visible nor reusable by other users until they are published on SpagoBI Server.

To get documents published on the server, configure SpagoBI Meta and connect it to SpagoBI Server, which will be the deployment environment to provide end-users with all certified objects. The main steps to configure SpagoBI Meta are:

- select the SpagoBI perspective
- create a SpagoBI project
- set SpagoBI Server connections
- set data sources.

At first execution of SpagoBI Meta, a welcome page appears with references to technical documentation (the wiki⁹, SpagoBI free community tool) and the preferred working perspective.

⁹ <http://wiki.spagobi.org/xwiki/bin/view/Main/>

BUSINESS INTELLIGENCE WITH SPAGOBI



FIGURE 3.3 – SpagoBI Meta welcome page

The first step is to create a SpagoBI project by selecting the SpagoBI icon from the toolbar located at the top of the page.

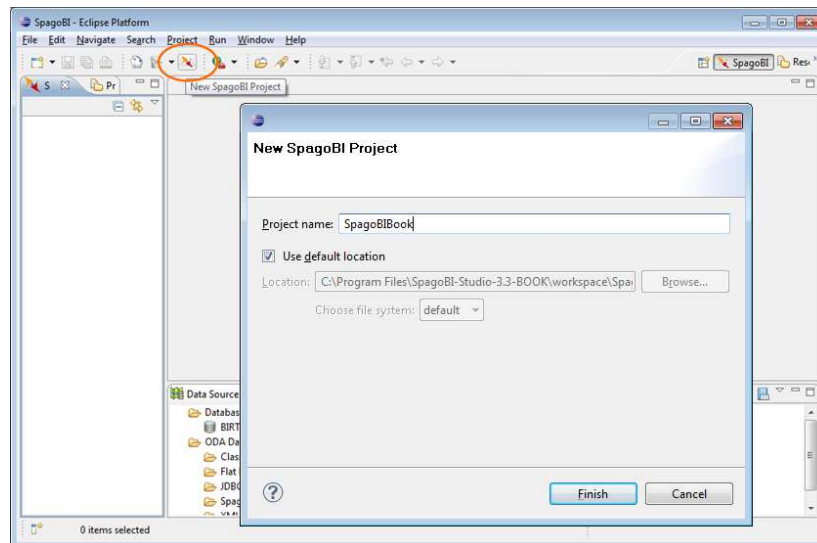


FIGURE 3.4 – Create a SpagoBI project

Once the project has been created, a standard folder tree appears.

The predefined folders are:

- **Business Models:** to host all metadata models created by the user
- **Business Queries:** to host all queries defined by the user over a metadata model
- **Resources:** to define all technical resources to be used in the project (i.e., the reference to SpagoBI Server)
- **Private folders:** to host personal or project documentation that may be useful for the user during his work. This folder can be freely managed by the user.

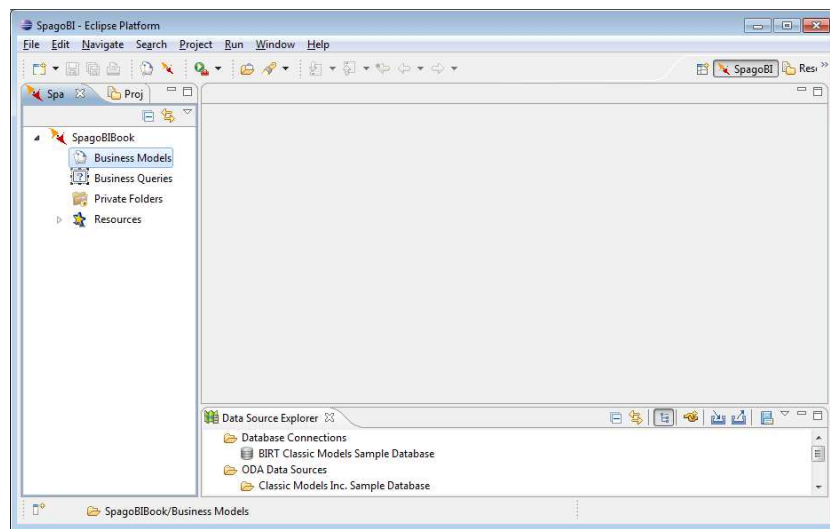


FIGURE 3.5 – SpagoBI Meta folders tree

Now you can configure references to one or more SpagoBI Servers. In other words, each user can work for many projects that can be:

- hosted on different servers
- hosted on the same server with different user accounts.

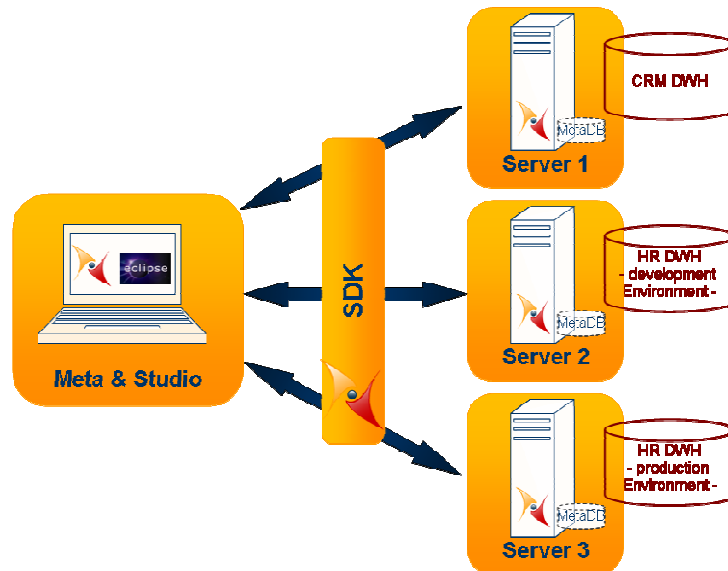


FIGURE 3.6 – SpagoBI Meta towards SpagoBI Servers

To define a server connection, click on the **New Server** item of the **Server** contextual menu. You will be asked for:

- **Server name:** a logical name to identify the server. The name is used on the local workspace only and has no relation with the physical one
- **Url:** the http URL where the server is hosted and reachable.
- **User:** the user who authenticates on SpagoBI Server, setting his access rights in terms of what kind of operations he can do (upload and download a model or a data set) and what parts of the Server repository he can access.
- **Password:** user's password
- **Active:** a flag that indicates the active server. It is particularly useful when the user is working with multiple servers. The active server indicates that every upload and download operation refers to this SpagoBI Server instance.

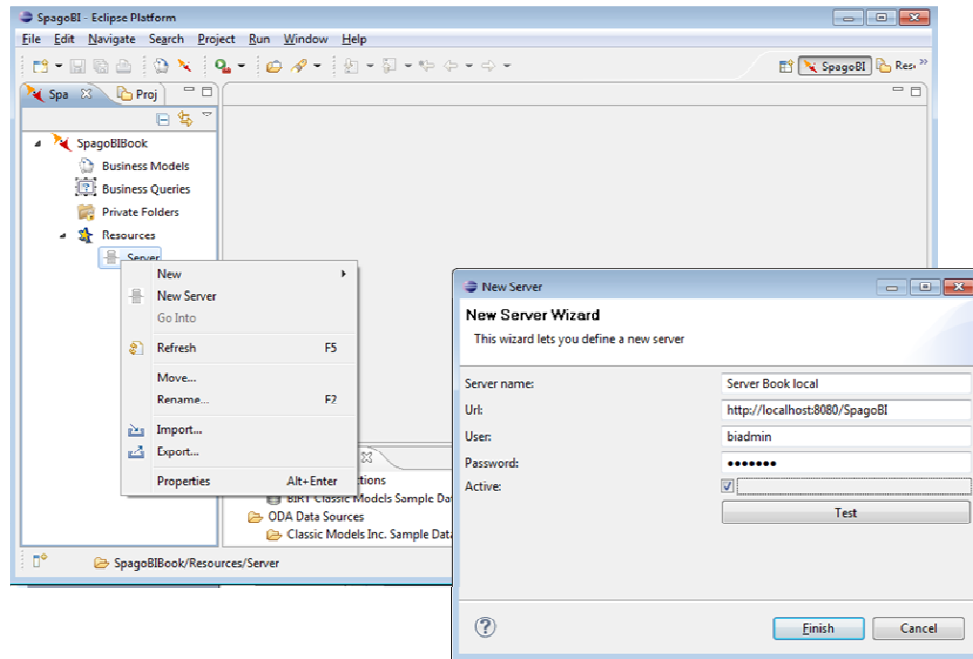
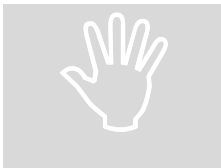


FIGURE 3.7 – Connection to a SpagoBI Server from SpagoBI Meta



Connection to SpagoBI Server

If something in your network configuration has been changed from your first run of SpagoBI Meta, the connection test of SpagoBI Meta to the Server could fail. Most often this problem is due to the proxy settings in your Eclipse environment. If this is not the case, try to run SpagoBI Meta from the command line with the `–clean` option to reset working settings. (`SpagoBI.exe –clean`)

The last step consists in setting data sources. As for servers, it is possible to define one or more data source connections to work with multiple metadata models that refer to different data sources.

At present, SpagoBI manages metamodels based on RDBMS only. Therefore, database connections are the only supported data sources.



DWH

From a technical point of view, nothing prevents you from using an operational database (OLTP) as data source. However this is not recommended. A DWH is preferred to guarantee performances, consistency and stability.

The user can refer to a local database or a remote one, depending on the general project resources. Obviously, if the user refers to a local database, this must be a copy (in terms of schema, not necessarily data) of the one referred by SpagoBI Server on which the metamodel will be deployed.

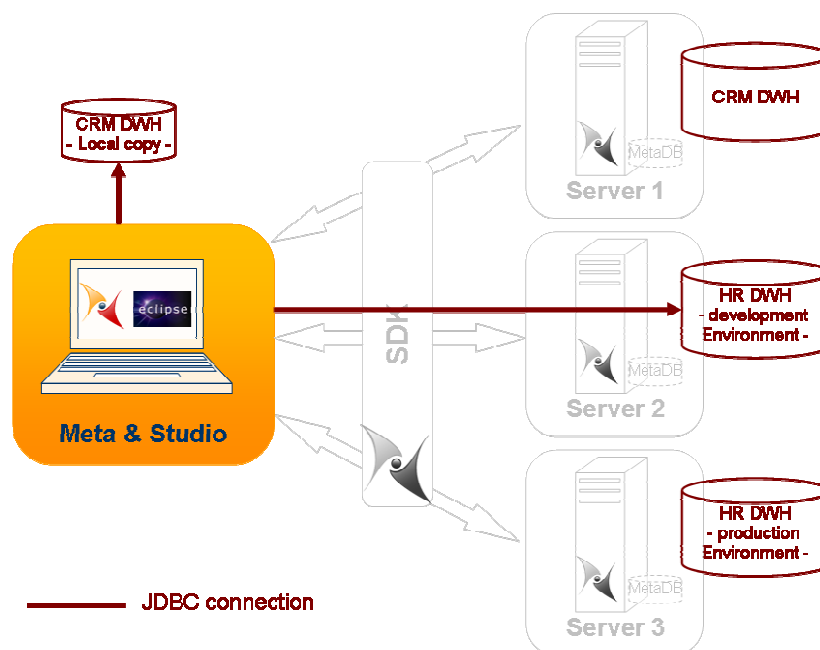


FIGURE 3.8 – Data sources on SpagoBI Meta

The procedure to define a data base connection follows the Eclipse standard procedure:

- Select the **New** item on the contextual menu of the **Data Base Connection** in the **Data Source Explorer** panel

BUSINESS INTELLIGENCE WITH SPAGOBI

- Create a connection profile choosing the RDBMS type and giving it a name
- Choose the right JDBC driver or add a new one
- Set connection parameters
- Test and save the new connection profile.

In the following, we show step by step how to create a data base connection.

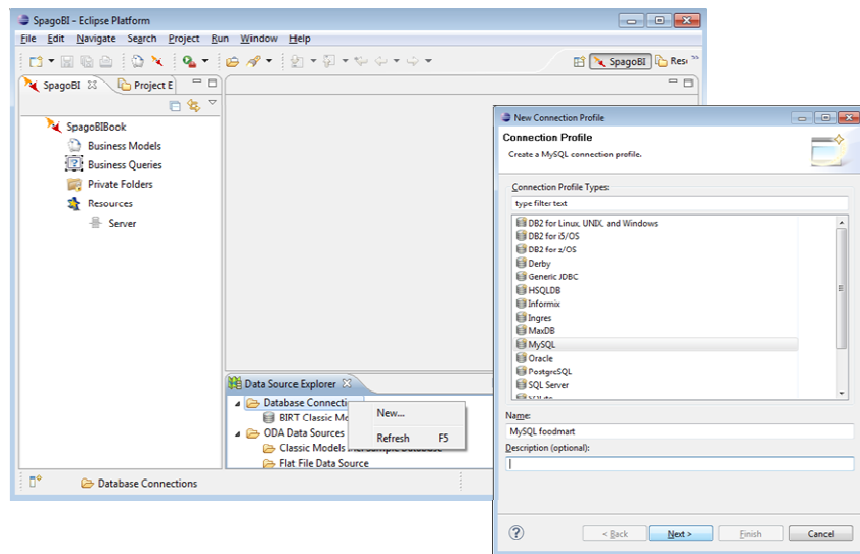


FIGURE 3.9 – Step 1 and 2 : Create a new connection profile selecting the right RDBMS type

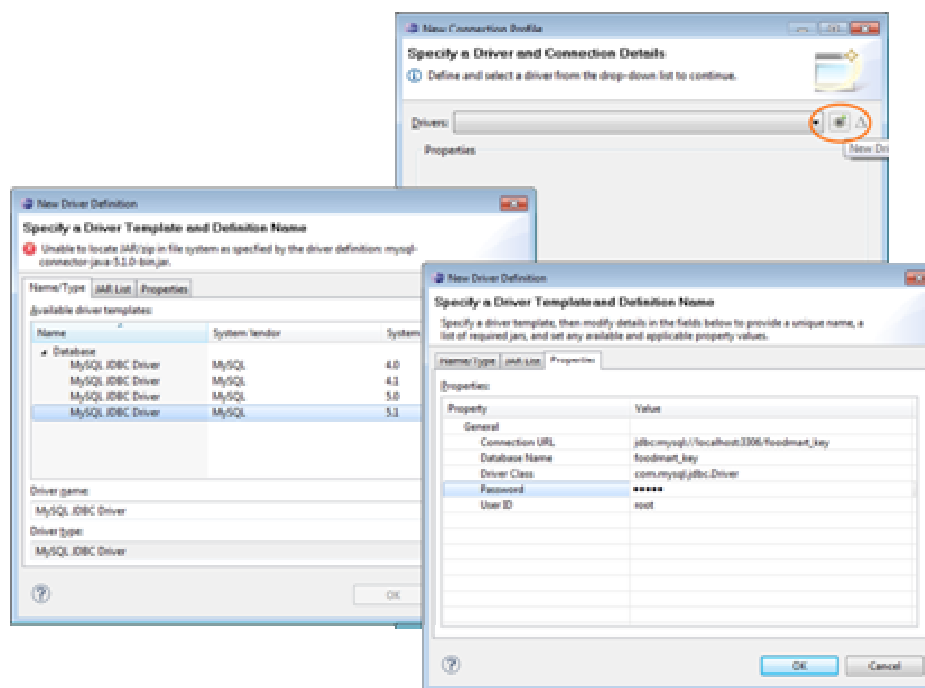


FIGURE 3.10 – Step 3 and 4 : Choose or add the right JDBC driver and set its properties

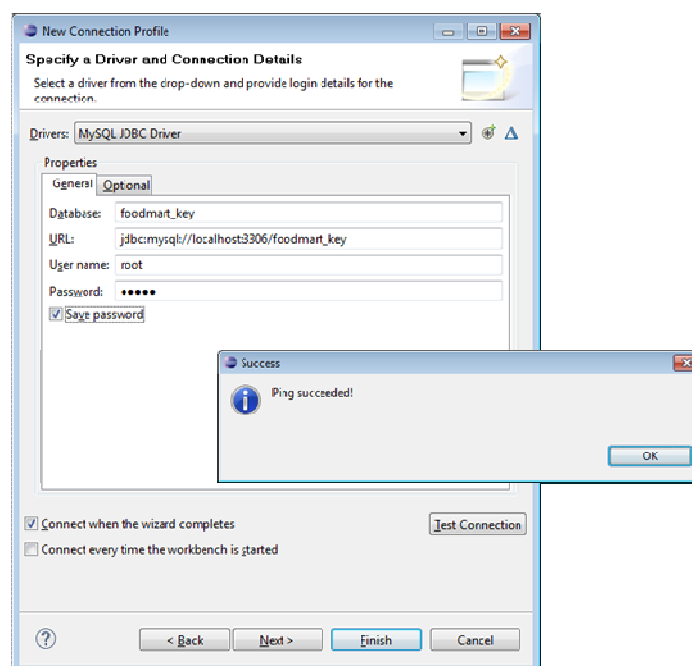


FIGURE 3.11 – Step 5 : Test and save the new connection profile

At this point, SpagoBI Meta is ready to work!

The business model

The *Business Model* (hereafter BM) is the semantic layer built over the data warehouse to redraw the data domain in terms of business elements. This allows to model and access data from the viewpoint of the end-user instead of the DBA.

The definition of a BM does not create new data structures, but produces a mapping between the business view and the technical one. Basically the mapping takes place at two different levels of abstraction, which correspond to the so called *physical model* and the business model itself.

The physical model is generated by choosing some (or all) tables from the data warehouse. At this level we have a representation of the data warehouse structure. The physical model defines the columns belonging to each table, the data type and constraints present within a column, as well as the (primary and foreign) keys and their relationships, if any. In short, the physical model shows data as they are organized in the data warehouse.

The next level of abstraction is the business model. The BM consists of business classes (notice that we do not refer to tables anymore). Classes may have attributes, as well as relationships with other classes. Business classes represent concepts or perspectives on data that do not necessarily correspond to existing structures in the data warehouse. The business model is indeed built by mapping its classes to the physical model.



The business model abstraction

Decoupling business data modeling from database design provides business users with a customized, transparent view on the data of interest, without affecting the physical model or the structure of the underlying database.

6. Analytical engines

In the previous chapter we learnt that the *analytical model* includes all the developed analysis of a BI project. We also learnt that SpagoBI supports several analysis domains (report, chart, cockpits, etc.) providing many *analytical engines* for each of them.

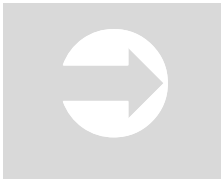
More in detail, SpagoBI Server provides:

- four engines for reporting
- three engines for multi-dimensional analysis
- three engines for charts
- one engine for interactive cockpits
- one engine for KPIs
- two engines for data mining
- two engines for free inquiry and driven data-selection
- one engine for ad hoc reporting
- two engines for location intelligence
- two engines for RT dashboards and consoles
- four engines for mobile
- one engine for office automation
- one engine for collaborative tools.

If you only select the dataset when creating the document, the report will execute by retrieving data from the selected dataset. The use of the data source depends on whether the parameter **connectionName** has a value: if this is the case, its value is used as a data source. If not, the data source is the one associated to the dataset on the Server.

Finally, you can leave the dataset blank when creating the document. At execution time, you will need to store a string describing the SQL query in the **query** parameter. Note that this parameter only accepts SQL syntax. The data source will be selected with the same rule as above.

Setting the dataset and the data source using parameters allows to associate them dynamically to the document. It is the natural option for an accessible table that is accessed from a stand-alone application.



Associate parameters to analytical documents

For a full understanding of parameters and how to associate them to documents, please refer to the Behavioral Model section in chapter 5 – SpagoBI Server.

Multidimensional analysis

Multidimensional analysis (OLAP) allows the hierarchical inquiry of numerical measures, over predefined dimensions. The user can monitor data on different detail levels and from different perspectives, through drill-down, drill-across, slice-and-dice, drill-through processes.

The main characteristics of an OLAP are:

- the need for a specific data structure (logical or physical)
- analysis based on dimensions, hierarchies and measures
- interactive analysis
- freedom to re-orient analysis

- different levels of data analysis, through synthetic and detailed views
- drill-down, slice and dice, drill-through.

For these reasons, OLAP usually has a high level of usage, mainly by analysts. In fact, it allows the selective navigation over data, through synthetic and detailed views, with a medium level of difficulty.

SpagoBI provides different OLAP engines:

- SpagoBIJPivotEngine, integrating the well-known JPivot/Mondrian OLAP engine
- SpagoBIJPaloEngine, integrating JPalo client over the well-known Mondrian server
- SpagoBIJXMLAEngine: a specific engine that uses the XMLA standard for a generic access to an OLAP server (such as MS Analysis services or SAP infocubes) using JPivot client.

SpagoBIJPivotEngine

SpagoBIJPivotEngine integrates Mondrian OLAP server and JPivot cube navigation client, to provide multi-dimensional analysis. These two modules are integrated within the same web application.

Mondrian is a *Relational Online Analytical Processing* (ROLAP) tool. It provides the back-end support for both SpagoBIJPivotEngine and SpagoBIJPaloEngine. OLAP structures, such as cubes, dimensions and attributes, are mapped directly onto tables and columns of the data warehouse. This way, Mondrian builds an OLAP cube in memory that can be accessed by client applications.

JPivot is the front-end tool to interact with Mondrian server and shows the results via the typical OLAP functionalities, e.g., drill down, slicing and dicing on a multi-dimensional table. JPivot translates user's navigation actions into MDX queries on the multi-dimensional cube, and shows query results on the table he is navigating.

In addition, JPivot allows the configuration of the navigation toolbar, the definition of profiling logics and the setting of cross navigation links, as we will see in the remainder of this section.

The creation of an OLAP analytical document with SpagoBIJPivotEngine requires the following steps:

- Cube modelling
- Catalogue configuration
- Template building
- Analytical document creation.

Note that the first two steps are common to both JPivot and JPalo Engine, and they are typical of OLAP engines.

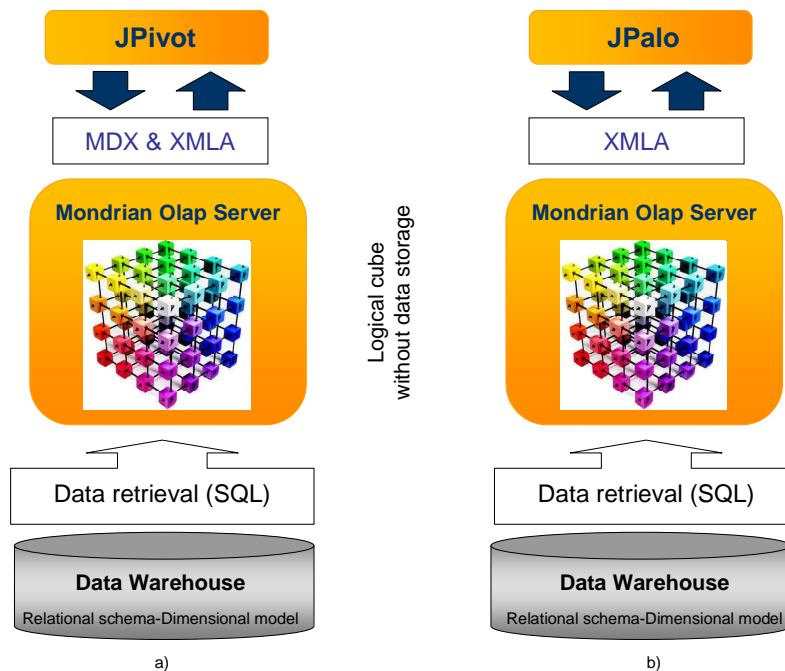


FIGURE 6.2 - Architecture of : a) JPivot and b) JPalo OLAP Engines

Creating the analysis with Mondrian OLAP Server

In this section we will go through the steps performed on Mondrian, namely:

- Cube modelling
- Catalogue configuration.

These are characteristic of the OLAP engine and they are shared by both JPivot and JPalo Engines.

Cube modelling

The very first step for a multi-dimensional analysis is to identify essential information describing the process/event under analysis and to consider how it is stored and organized in the database. On the basis of these two elements, a mapping process should be performed to create the multi-dimensional model.



From the relational to the multi-dimensional model

The logical structure of the database has an impact on the mapping approach to be adopted when creating the multi-dimensional model, as well as on query performances.

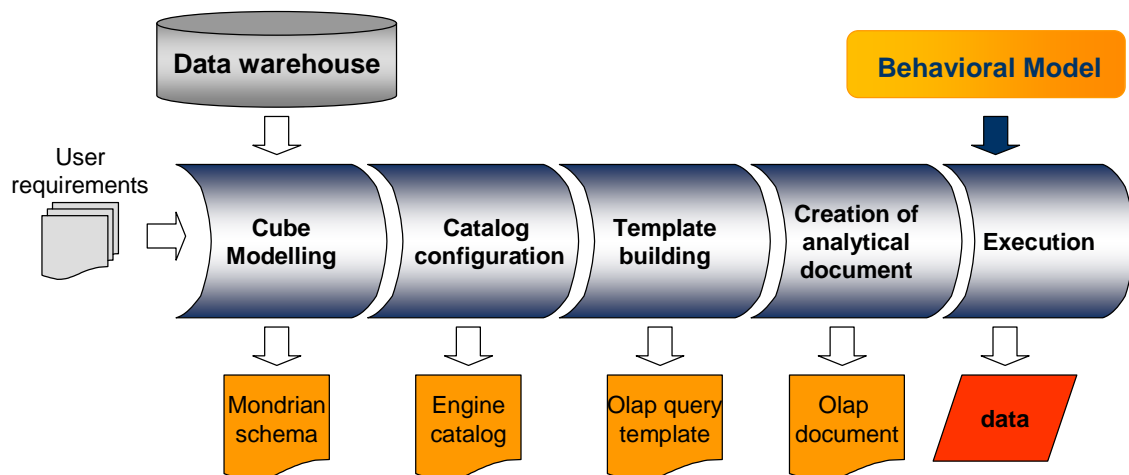


FIGURE 6.3 - Creation of an OLAP analytical document

If the structure of the relational schema complies with multi-dimensional logics, it will be easier to map the entities of the physical model onto the metadata used in Mondrian schemas. Otherwise, if the structure is highly normalized and scarcely dimensional, the mapping process will probably require to force and approximate the model to obtain a multi-dimensional model.

As said above, Mondrian is a ROLAP tool. As such, it maps OLAP structures, such as cubes, dimensions and attributes directly on tables and columns of a relational database via XML-based files, called Mondrian *schemas*. Mondrian schemas are treated by SpagoBI as resources and organized into catalogues.

Hereafter, an example of Mondrian schema.

```
<?xml version="1.0"?>
<Schema name="FoodMart">

  <!-- Shared dimensions -->
  <Dimension name="Customers">
    <Hierarchy hasAll="true" allMemberName="All Customers"
primaryKey="customer_id">
      <Table name="customer"/>
      <Level name="Country" column="country"
uniqueMembers="true"/>
      <Level name="State Province" column="state_province"
uniqueMembers="true"/>
      <Level name="City" column="city"
uniqueMembers="false"/>
    </Hierarchy>
      ... Other hierarchies
  </Dimension>
    ...
  <Cube name="Sales">
    <Table name="sales_fact_1998"/>
    <DimensionUsage name="Customers" source="Customers"
foreignKey="customer_id"/>
    ...
  <!-- Private dimensions -->
  <Dimension name="Promotion Media" foreignKey="promotion_id">
    <Hierarchy hasAll="true" allMemberName="All Media"
primaryKey="promotion_id">
      <Table name="promotion"/>
      <Level name="Media Type" column="media_type"
uniqueMembers="true"/>
    </Hierarchy>
  </Dimension>
    ...
```

```

<!-- basic measures-->
<Measure name="Unit Sales"    column="unit_sales"
aggregator="sum"    formatString="#,###.00"/>
<Measure name="Store Cost"    column="store_cost"
aggregator="sum"    formatString="#,###.00"/>
<Measure name="Store Sales"    column="store_sales"
aggregator="sum"    formatString="#,###.00"/>

<!-- derived measures-->
<CalculatedMember name="Profit" dimension="Measures">
    <Formula>
        [Measures].[Store Sales] - [Measures].[Store Cost]
    </Formula>
<CalculatedMemberProperty name="format_string"
value="$#,##0.00"/>
    </CalculatedMember>
</Cube>
...
</Schema>

```

Each mapping file contains one schema only, as well as multiple dimensions and cubes. Cubes include multiple dimensions and measures. Dimensions include multiple hierarchies and levels. Measures can be either primitive, i.e., bound to single columns of the fact table, or calculated, i.e., derived from calculation formulas that are defined in the schema.

The schema also contains links between the elements of the OLAP model and the entities of the physical model: for example, <table> sets a link between a cube and its dimensions, while the attributes `primaryKey` and `foreignKey` reference integrity constraints of the star schema.



Mondrian

For a detailed explanation of Mondrian schemas, please refer to the documentation available at the official project webpage: <http://mondrian.pentaho.com/>

Engine catalogue configuration

To reference an OLAP cube, first insert the corresponding Mondrian schema into the catalogue of schemas managed by the engine. When creating a new

template, you can choose among the available cubes using their registered schemas.

- Put a copy of the .XML schema file into the folder /resources/olap of SpagoBI Server.
- Open the configuration file engine-config.xml, located at <your_root_installation>\webapps\SpagoBIJPivotEngine\WEB-INF\classes and add your schema as in the example below.

```
<Engine_configuration>
    ...
    <Schemas>
        <schema catalogUri="/Olap/schema_name.xml"
                name="schema_logical_name"
        /
    </ Schemas >
</ Engine_configuration >
```

Where:

- **catalogURI** is the name of the .XML file containing the schema definition that you have previously copied in /resources/olap
- **name** is the actual name of the schema.

Creating and executing the analytical document with JPivot Client

Template building

Once you have created the cube, you need to build the mapping template between the cube and the analytical document. There are two options for building an OLAP mapping template in SpagoBI JPivotEngine.

- Manually edit the template. This option is recommended to very expert users only.
- Use the template building wizard provided during the development of the OLAP analytical document.

In the following discussion we will adopt the second approach. Then we will shortly illustrate the structure of the mapping template and give hints for manual editing.

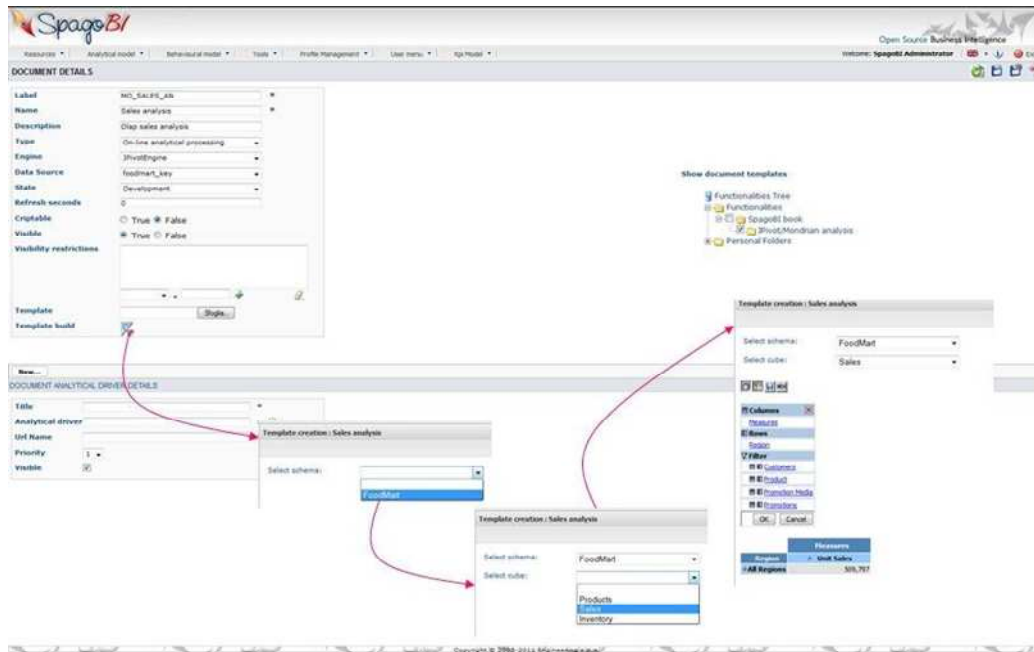



FIGURE 6.4 - JJPivot template building wizard of an OLAP query

We are going to follow these steps:

- Open the wizard clicking on the small icon at the bottom 
- Select the desired schema among those registered on the server
- Select a cube within the chosen schema.

Here the wizard will compose a default MDX query and draw the corresponding pivot table. Most likely you will need to modify the default visualization, so the last step is:

- Configure the default visualization until the searched result is reached.

The visualization can be modified in two ways:

- Edit the MDX query or
- Edit the pivot table using the **Cube Configurator**.

Any Change performed on a table will affect the other one and vice versa.

We will first go through the query editor. The features of the **Cube Configurator** are shown in section JPivot Client.

MDX Query Editor

When clicking on the MDX button, a window opens where you can both edit the query and add parameters.

The query is written using the Multi Dimensional eXpression (MDX) language. MDX is a specific language to operate on data structures managed by OLAP systems. MDX queries cannot be managed by traditional SQL clients: specific OLAP clients are needed to handle MDX queries and results.

Here is the simple MDX query from the example above:

```
select {[Measures].[Unit Sales]} on columns,  
       {[Region].[All Regions]} on rows  
from [Sales]  
where [Product].[All Products].[Drink]
```



MDX Syntax specification

To learn more about the MDX language specification and its usage, please refer to online documentation.

<http://msdn.microsoft.com/en-us/library/ms145506.aspx>

In the MDX editor window you can also associate parameters to the query. This is similar to the association of parameters to a dataset defined as a SQL query. If you insert a parameter with a given URI into the MDX query, remember to associate the same parameter URI to the OLAP analytical document on the Server. This way, the parameter can be associated to an analytical driver.



Analytical Drivers

Read how to associate analytical drivers to documents at section Behavioral Model, in chapter 5 - SpagoBI Server.

Once you are happy with the result, save the template by clicking on the corresponding icon.

Mapping template structure

The template generated by the wizard is an .XML file telling SpagoBI JPivotEngine how to navigate the OLAP cube.

As said above, very expert users can edit the template by hand. The following is an example of mapping template:

```
<?xml version="1.0" encoding="UTF-8"?>
<olap>
    /* schema configuration*/
    <cube reference="schema_name"/>
    /* query configuration*/
    <MDXquery>
        ...
    </MDXquery>
    <MDXMondrianQuery>
        ...
    </MDXMondrianQuery>
    /* toolbar configuration*/
    <TOOLBAR>
        ...
    </TOOLBAR>
    /* cross navigation configuration*/
    <CROSS_NAVIGATION    >
        ...
    </CROSS_NAVIGATION>
    /* data profiling*/
    <DATA-ACCESS>
        ...
    </DATA-ACCESS>
</olap>
```


The different sections of the file follow:

- The `CUBE` section sets the Mondrian schema. It should reference the exact name of the schema, as registered in the catalogue on the Server
- The `MDXMondrianQuery` section contains the original MDX query (the one edited via the wizard)
- The `MDX` section contains a variation of the original MDX query, as used by SpagoBI Engine. This version includes parameters (if any).

Below the details of the MDX query sections. The name of the parameter will allow SpagoBI to link the analytical driver associated to the document via the parameter (on the Server).

```
<?xml version="1.0" encoding="UTF-8"?>
<olap>
/* schema configuration*/
<cube reference=" FoodMart"/>
/* query configuration*/
  <MDXquery>
    select          {[Measures].[Unit Sales]} ON COLUMNS,
                   {[Region].[All Regions]} ON ROWS
    from [Sales]
    where [Product].[All Products].[ ${prodFam}]
    <parameter name="productFamily" as="prodFam"/>
  </MDXquery>

  <MDXMondrianQuery>
    select {[Measures].[Unit Sales]} ON COLUMNS,
           {[Region].[All Regions]} ON ROWS
    from [Sales]
    where [Product].[All Products].[Drink]
  </MDXMondrianQuery>
</olap>
```

- The `toolbar` section is used to configure visibility options for the toolbar in the OLAP document.

Below an example of template excerpt, where the various buttons are listed. Some of them are visible (i.e., the attribute `visible` is set to `true`) while others are not (the attribute is set to `false`).

The exact meaning and functionalities of each toolbar button are explained in the next section, JPivot Client.

```
<?xml version="1.0" encoding="UTF-8"?>
<olap>
    ...
    /* toolbar configuration*/
    <TOOLBAR>
        <BUTTON_CUBE visible="true" />
        <BUTTON_MDX visible="false" />
        <BUTTON_ORDER visible="false" />
        <BUTTON_FATHER_MEMBERS visible="true" />
        <BUTTON_HIDE_SPANS visible="false" />
        <BUTTON_SHOW_PROPERTIES visible="false" />
        <BUTTON_HIDE_EMPTY visible="true" />
        <BUTTON_SHIFT_AXIS visible="true" />
        <BUTTON_DRILL_MEMBER visible="true" />
        <BUTTON_DRILL_POSITION visible="true" />
        <BUTTON_DRILL_REPLACE visible="false" />
        <BUTTON_DRILL_THROUGH visible="false" />
        <BUTTON_SHOW_CHART visible="false" />
        <BUTTON_CONFIGURE_CHART visible="false" />
        <BUTTON_CONFIGURE_PRINT visible="false" />
        <BUTTON_FLUSH_CACHE visible="false" />
        <BUTTON_SAVE visible="true" />
    </TOOLBAR>
    ...
</olap>
```

Creating the analytical document

As usual in SpagoBI, the analytical document should be created on the Server. Once you have the template ready (either produced via the wizard or another tool, or manually edited), you can create the OLAP document on the Server.

To create a new OLAP document, create a new document in the Analytical Documents area, then select **Online analytical processing > JPivot Engine**. Choose a name, a functionality, load the mapping template and save. You will see the document in the functionality (folder) you selected, displayed with the typical cube icon (see FIGURE 6.5).

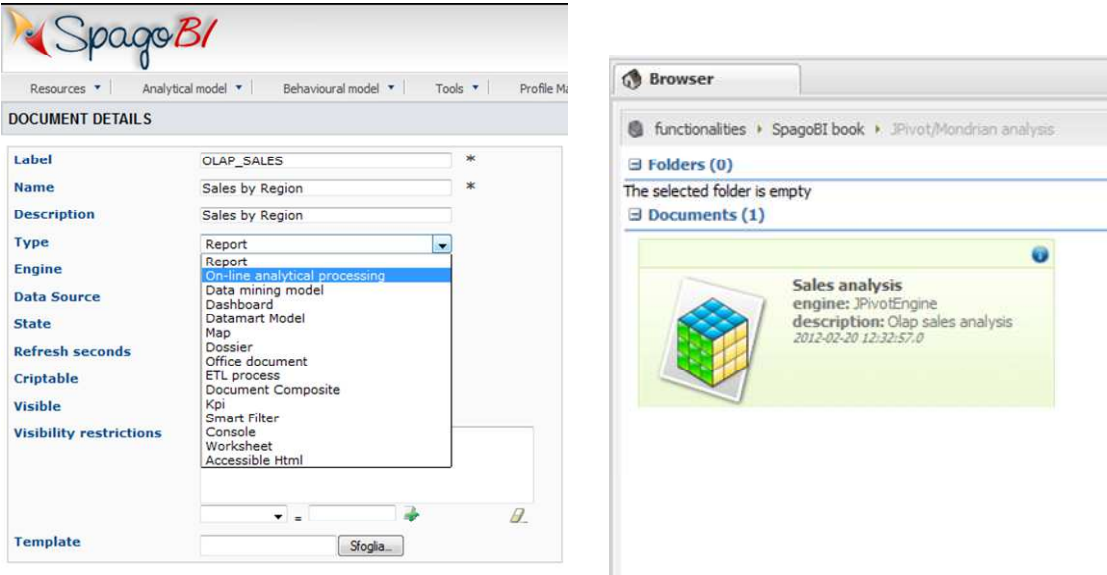


FIGURE 6.5 – Creation of an OLAP document

Executing the OLAP document

OLAP analytical documents created with SpagoBI JPivotEngine can be navigated thanks to the JPivot client. To run the JPivot client, just open the OLAP document.

JPivot provides a toolbar with all typical navigation functionalities of an OLAP navigation, such as drill (down, through and across), slicing & dicing, navigation view configuration, charts generation and export of results in various formats.

TABLE 6.1 summarizes the meaning of each item contained in the toolbar.



Toolbar customization

Since the navigation functionalities require expertise to be properly used, it may be wise to customize the toolbar according to the user's level of knowledge. This can be done by editing the mapping template (see above).