TECHNISCHE UNIVERSITÄT DRESDEN

Department of Business Management and Economics
Chair of Business Informatics, esp. Information Systems in
Manufacturing and Commerce

Diploma Thesis

Automating ERP Package Configuration for Small Businesses

Name:	Klaus Wölfel
Address:	Quellenstr. 6, 97204 Höchberg, Germany
Registration Number:	3050928
Submitted to:	Prof. Dr. Susanne Strahringer
Submission Date:	July 21, 2010

Restricted Note

This paper remains restricted to the public by virtue of confidential data and information.

Contents

List of Abbreviations

BOM	Bill of Material
CEO	Chief Executive Officer
CRM	Customer Relationship Management
DMS	Document Management System
EAT	ERP5 Artificial Intelligence Toolkit
ECOWAS	Economic Community of West African States
ERP	Enterprise Resource Planning
EU	European Union
IETF	Internet Engineering Task Force
IS	Information System
ISV	Independent Software Vendor
IT	Information Technology
MRP	Material Requirements Planning
MRP II	Manufacturing Resource Planning
OSOE	One Student One ERP
PDM	Product Data Management
SaaS	Software as a Service
SCM	Supply Chain Management
SER	SIP Express Router
SME	Small and Medium Enterprise
SQL	Structured Query Language

TIO	Total Information Outsourcing
UBM	Unified Business Model
VCS	Version Control System
VoIP	Voice over IP
XML	Extensible Markup Language
ZEO	Zope Enterprise Objects
ZMI	Zope Management Interface
ZODB	Zope Object Data Base
ZPT	Zope Page Templates

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Enterprise Resource Planning (ERP) systems are said to enable organizations to manage their resources efficiently and effectively by providing a total and integrated solution for their information processing needs (Nah, Lau, & Kuang, 2001, p. 285). Due to technical and economical restrictions, ERP systems traditionally have been focused on larger organizations. In recent years however, a turn of the market towards Small and Medium Enterprises (SMEs) can be observed (Deep, Guttridge, Dani, & Burns, 2008, p. 431). Adam and O'Doherty (2000, p. 314) show that SMEs are as likely to be interested in ERP as multinational organizations. ERP packages are being viewed as a key factor for gaining competitive advantage in the SME sector and empirical findings confirm these expectations (Koh & Simpson, 2007, p. 73).

However, Morabito, Pace, and Previtali (2005, p. 591) identify lack of human and financial resources as well as lock-in risks as major problems that SMEs face when adopting ERP technology. They often do not have dedicated teams for implementation and software maintenance and cannot spend as much money on Information Technology (IT) as large enterprises, which in turn makes them more vulnerable to the risk of lock-ins in ERP packages when requirements change after implementation.

Business models, where SMEs access ERP functionalities through the Internet instead of purchasing them could alleviate these problems and broaden the ERP market (Adam & O'Doherty, 2000, p. 305). Recently Software as a Service (SaaS) is associated to this kind of business model (Hofmann, 2008, p. 87). By providing applications directly through the Internet, SaaS eliminates installation and update tasks, thus saving clients from maintenance work and reducing IT expenses by on-demand pricing (Wang et al., 2008, p. 827).

Another "disruptive business model" mentioned by Hofmann (2008, p. 87) is that of open source companies. Free / open source ERP systems might be an alternative for SMEs as they tackle their specific problems. They not only help to save license costs, but they also prevent lock-in. As their source code is free to everyone they lower the barrier for third parties to perform modifications. (Campos, Carvalho, & Rodrigues, 2007, p. 3).

Despite these promising perspectives the consulting required to implement an ERP system remains a financial burden (Janssens, Kusters, & Heemstra, 2007, p. 23). Although ERP systems are cheaper and easier to implement for SMEs than for large enterprises (Morabito et al., 2005, p. 591), SMEs may face challenges in affording major consulting support (Snider, Da Silveira, & Balakrishnan, 2009, p. 50; Kinni, 1995, p. 50). Implementation costs are often far exceeding the costs for ERP package licenses. Thus the greatest savings can be achieved during implementation (Timbrell & Gable, 2002, p. 1117).

Off-the-shelf ERP packages are implemented mainly by configuration (Brehm, Heinzl, & Markus, 2001, p. 1). The author believes, that automating this configuration process would lessen the burden of the implementation process and make ERP more accessible for SMEs. The vision is that a packaged ERP system will be automatically configured based on a questionnaire filled out by the Chief Executive Officer (CEO) of a small business. The first example of such automation is the SaaS “TioLive” which uses various wizards to automate the configuration of the open source ERP system “ERP5”.

However, current technology is still very simple. To further pursue this vision, two approaches for automating the configuration of packaged ERP Software based on questionnaires are investigated: knowledge engineering with decision trees and classification based on machine learning algorithms. Applying these approaches to ERP5 will make those wizards more intelligent and will allow to provide a solution which matches the requirements of a small business far better than before.

A successful application of the investigated approaches to the configuration options of ERP5 would be the foundation to create an automated system that can accomplish the bulk of the work needed to configure ERP5 for a specific adopting organization. In a SaaS-based setup, this basic configuration could then be refined by human IT consultants on demand over the Internet. Thus, the customer would experience the tailoring process of his ERP package as an integrated online service. Such a service could be called “Cloud Consulting”.

1.2 Research Design

The research objective is to investigate the automation of the adaption of a packaged ERP to the specific business needs of a SME. This is done on the basis of a specific open source ERP package, named ERP5 by Nexedi. Brehm et al. (2001, 4–6) call this adaption “tailoring” and identify different types of ERP package tailoring with different impact on the ERP system (see chapter 3: ERP Package Tailoring, p. 15). To reach the research objective, the following questions have to be answered:

- Which tailoring options are most likely suitable for automation generally and in the case of ERP5 specifically?
- How can these ERP5 tailoring options be automated?

The procedure to answer these questions is based on the design science paradigm. The idea is to better understand and solve human and organizational problems by creating innovative artifacts and applying them (Hevner, March, & Park, 2004, p. 75). The artifacts to be designed in this thesis are building blocks of an automated configuration system. The first type of artifacts consists in a decision tree that show how ERP parameters can be configured based on knowledge engineering. The second type of artifacts are prototypical code examples which feed classifiers with sample data

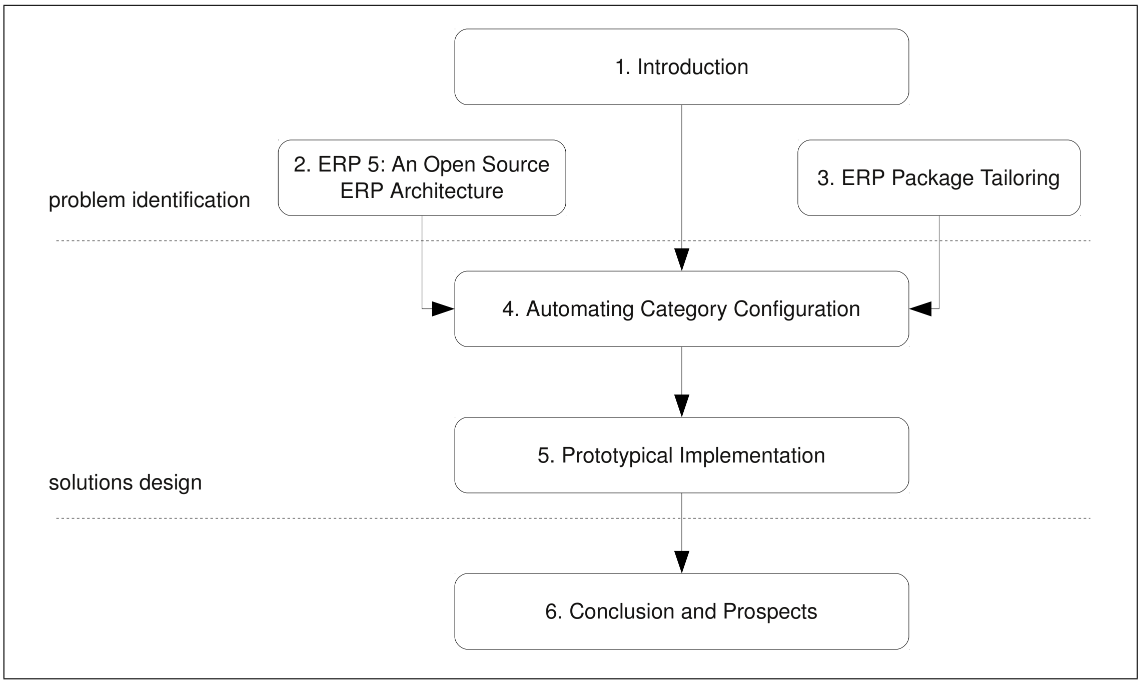

1. Introduction
2. ERP 5: An Open Source ERP Architecture
3. ERP Package Tailoring
problem identification
4. Automating Category Configuration
5. Prototypical Implementation
solutions design
6. Conclusion and Prospects

Figure 1: Structure of the thesis

to show how an ERP can be configured based on machine learning. The third artifact is a prototype for three ERP5 modules that help to create questionnaires and decision trees in ERP5. They can be used for data mining and form the basis for future automatic configuration. (see chapter 5: Prototypical Implementation, p. 49). Configuration use cases are applied to the designed artifacts to test the viability of the approaches.

The information required to implement these approaches is gathered through expert interviews, desk research, analyzing previous ERP5 configuration projects and an exemplary configuration case. The procedure to design the automation artifacts (chapter 4.1: Automation Procedure and Information Sources, p. 38) is roughly based on the design as a search process (Hevner et al., 2004, pp. 88–90). Chapters 1 to 3 are geared towards the problem identification phase and chapters 4 to 5 are oriented to the solution design phase (see figure 1). Within the scope of this thesis only the first steps of the design science process (Offermann, Levina, Schönherr, & Bub, 2009, pp. 4–5) are conducted.

1.3 Course of the Investigation

The structure of the thesis is aligned to the design as a search process and to the procedure of answering the previously outlined research questions. Figure 1 (Structure of the thesis, p. 3) shows, how the individual chapters depend on each other.

After an introduction to ERP5 and TioLive in chapter 2 (ERP5: An Open Source ERP Project, p. 5), the first research question is discussed in chapter 3 (ERP Package Tailoring, p. 15). A typology of tailoring options by Brehm et al. (2001, pp. 4–6) is introduced before examining the tailoring types for their applicability to automation. The chapter continues with a presentation of tailoring options in ERP5. It focuses on the configuration of categories which are the target of the automation approaches investigated in this thesis. Finally, the configuration case of a small company which wants to adopt TioLive is presented. Its category configuration is later used as informational input to the design of the automation artifacts.

Chapter 4 and 5 are meant to answer the second and third of the research questions in discussing two automation approaches and designing an exemplary implementation for ERP5 from which requirements for an applicability of the approaches are deduced. Chapter 4 (Automating Category Configuration, p. 38) explains the procedure to develop the automation of ERP5 category configuration including how and from which sources the needed information is acquired. Two automation approaches are discussed, decision trees and data mining. Their applicability to automate the configuration of selected ERP5 categories is analyzed. The chapter concludes with a proposal to gather and evaluate possible questions for the configuration questionnaire as well as to discover their interdependencies.

The designed decision tree and the documentation of the design procedure are presented in chapter 5 (Prototypical Implementation, p. 49). The viability of the decision tree is evaluated through a configuration use case and previous ERP5 implementation projects. For the machine learning approach, two prototypical examples are developed, one for questions with free text replies an one for replies with discrete values. Furthermore, a prototype is presented which consists of three ERP5 modules that allow to create questions of different types, build questionnaires and decision trees, display them and collect answers.

The thesis concludes with chapter 6 (Conclusion and Outlook, p. 58). It summarizes the achieved results in terms of the application of the investigated approaches to the automation of ERP5 category configuration. The course for further investigations is outlined to improve the designed artifacts into a delivery state and to apply the approaches to other ERP5 configuration options.

2 ERP5: An Open Source ERP Project

2.1 Introduction

ERP5 is a free / open source ERP project, born in 2001 at the initiative of two French companies, Nexedi and Coramy. Nexedi is the main developer of ERP5 which was first deployed at Coramy, an apparel producer (Smets-Solanes, 2002). Since then it is developed and used by a growing international community from France, Brazil, Germany, India, Japan, Mongolia, Poland and Senegal among others (Monnerat, Carvalho, & Campos, 2008, p. 1063; Nexedi SA, n.d.; Honoré & Smets, 2010). ERP5 targets SMEs as well as larger organizations (Nexedi SA, n.d.). Apart from apparel, it has been deployed in various industries, among them aerospace, automotive, e-commerce, software service companies, a central bank, a hospital and a government agency (Smets, 2008).

The ERP5 project is not only producing open source code, but also open documentation and open educational material which is freely available at the ERP5 knowledge base (`http://www.myerp5.com/kb`) and the ERP5 wiki (`http://www.erp5.org`). Smets-Solanes and Carvalho (2003) have published the architecture of ERP5 and its underlying theoretical model. This theoretical foundation in conjunction with the accessibility of technical and practical information makes ERP5 interesting for Information System (IS) research.

2.2 General Goals of ERP

ERP systems are comprehensive software applications that aim to integrate organizational processes through shared information and data flows (Shanks & Seddon, 2000, p. 243). They support most commercial functions of an organization including purchasing, inventory management, production planning, project management, finance, human resources and sales (Davenport, 1998, p. 122).

ERP systems have their roots in production planning and control techniques, namely Material Requirements Planning (MRP) and Manufacturing Resource Planning (MRP II) (Chen, 2001, pp. 375–376). Based on current and expected customer demands MRP calculates the quantities of parts and raw materials needed to produce the end items as well as the time periods when the components have to be produced or purchased depending on their inventories. Production and purchase orders are then generated based on these calculations. By including more business functions MRP II systems no longer consider material alone but also other resources like cash, labor and machines which enable them to not only predict a company's inventory but also the future of its liquid funds.

Today, ERP systems are capable of planning and managing virtually all resources of an organization. By taking into account more of the supply chain ERP systems strive to extend the scope of resource management from planning all *internal* resources to also plan and schedule *supplier* resources on *customer* demands and schedules. In line with developments in electronic business, support for electronic market places is becoming more important for ERP systems as well as deeper integration of external customers and suppliers through Customer Relationship Management (CRM) and Supply Chain Management (SCM) (Chen, 2001, p. 381–384; Shanks & Seddon, 2000, p. 243).

2.3 ERP5's Functional Scope

Functionality wise ERP5 fits well into the above description of ERP systems. Due to its first deployment in apparel industry, it has originally been designed as a production oriented ERP solution. As such it features classic ERP functions like Product Data Management (PDM), production planning and control with MRP, SCM, sale order and shipping. These are combined with finance and human resources functions such as accounting, invoicing, budgeting and pay roll. CRM is implemented by tracking customer relations (support requests, meetings, sales prospects). To support knowledge-intensive business services, ERP5 integrates a knowledge management system as well as a project management module. E-commerce is supported through a front end for online order management (online shop and e-procurement). It synchronizes with the back office server through the SyncML protocol so that a single products catalog can be shared by multiple vendors. (Smets-Solanes & Carvalho, 2003; Smets, 2007, p. 11).

All ERP5 functions are accessed through a web-based interface. A generic workflow engine is used to implement the supported business processes. What makes ERP5 fundamentally different from other ERP systems is:

- its abstract model which defines only five classes to represent all business processes within one and between multiple organizations (chapter 2.4.2: Unified Business Model, p. 9) and
- its document-centric approach to implement these business processes.

Other than a data- or process-centric paradigm, the document-centric approach focuses on the operational documents, their fields and document workflows. It assumes that every business process relies on a series of documents and that the architecture of an organization is discoverable through the list of operational documents which support this organization. The fields of the documents represent the data and their relations. The document-flow in a company corresponds to the workflows of its business processes (Atem de Carvalho & Monnerat, 2007, p. 339; Smets, n.d., p.6;).

The ERP5 core concepts are induced to a great extent by the technologies on which ERP5 is built upon. This means that for understanding ERP5, it helps to take a look at the underlying technological architecture which is also the basis for some of ERP5's specific features described later in this chapter.

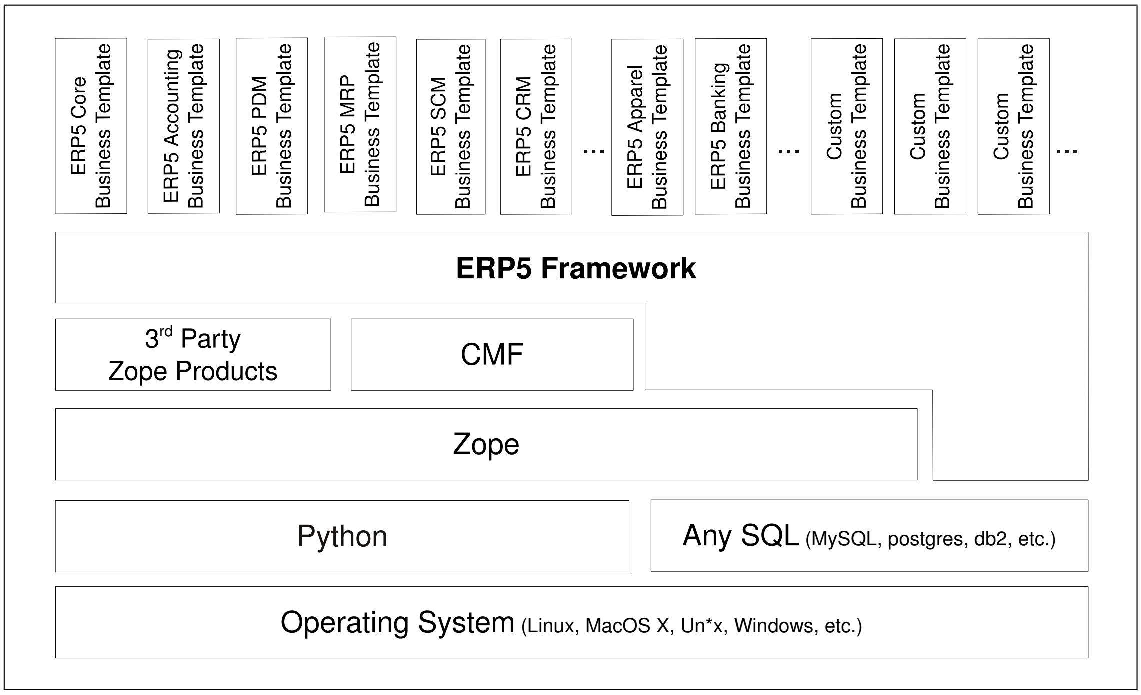

ERP5 Core Business Template
ERP5 Accounting Business Template
ERP5 PDM Business Template
ERP5 MRP Business Template
ERP5 SCM Business Template
ERP5 CRM Business Template
...
ERP5 Apparel Business Template
ERP5 Banking Business Template
...
Custom Business Template
Custom Business Template
Custom Business Template
...
ERP5 Framework
3rd Party Zope Products
CMF
Zope
Python
Any SQL (MySQL, postgres, db2, etc.)
Operating System (Linux, MacOS X, Un*x, Windows, etc.)

Figure 2: ERP5 technologies adapted from Smets (2007, p. 5)

2.4 ERP5 Architecture

2.4.1 Technical Components

One of the design goals of the ERP5 project is to use only one object-oriented programming language for the core system as well as for business logic and tailoring scripts. ERP5 is implemented in python, a dynamic high-level programming language. It allows code that has been originally written as tailoring scripts to be incorporated afterwards into core components. The second reason that lead to the choice for Python are its meta programming capabilities which allow to redefine the semantics of the implementation language at run time. ERP5 uses this technique to automatically generate most of its elementary methods which drastically reduces the number of lines of code. For example accessors for attributes and relations are all generated automatically from lists of properties (Smets, 2004).

Figure 2 gives an overview, how ERP5 technologies build on each other. The ERP5 Framework is based on the Zope Application Server (`http://www.zope.org`) which is also an open source system. It runs on multiple platforms, including Linux, Windows and MacOSX. Its core functionalities include web object publishing, role-based security for objects, user authentication and through-the-web development facilities.

For persistence of data, Zope includes the Zope Object Data Base (ZODB) which stores objects in a fully transactional way and keeps the transaction history for each object for undo functionality. Every object can be exported or imported in XML format. ERP5 improves on this capabilities with synchronization functionality that enables two or more ERP5 sites to share synchronized objects through the SyncML protocol (Atem de Carvalho & Monnerat, 2007, p. 341). Zope Enterprise Objects (ZEO) provides clustering and load balancing. Requests to one single Zope application can be distributed to multiple computers. NEO, a distributed storage engine for the ZODB developed during a research project in the ERP5 ecosystem extends the load balancing capabilities to the object store and adds full data replication over multiple machines. It is based on a peer-to-peer transaction protocol, that is in the process of being formally proved through model-checking (Bertrand et al., 2009, p. 315).

Zope also provides management of components, called Products. One of the add-on products used by ERP5 is the CMF (`http://www.zope.org/Products/CMF`), a Content Management Framework that allows the creation of document types and maintains them in a registry to provide services for documents such as associated actions, workflows, presentation and cataloging. ERP5 business objects are represented by documents of certain types. Rapid prototyping is possible as new document types can be created through-the-web based on existing types and associated with actions. Being based on Zope and the CMF means that web content management and e-business functions are not additions to ERP5, but at its core.

The CMF includes DCWorkflow, a general workflow engine that allows the design of workflows in the Zope Management Interface (ZMI). The workflow definition mainly consists of states and possible transitions that can occur, if certain conditions are met. If a workflow is associated with a document type, all documents of this type automatically get a state variable indicating their current state. They also acquire the actions defined for workflow transitions additionally to the document's actions. When a document is in a given state the authorized actors have a set of available workflow actions they can perform on the entity, depending on the conditions defined for the workflow transitions (Smets-Solanes & Carvalho, 2003, p. 40).

Actions can be python scripts, views on the document, dialog forms or a combination of these. Zope Page Templates (ZPT) provide rapid implementation of web user interface presentation logic based on XML. Form views and dialogs can be assembled of widgets in a web-based point-and-click interface which is provided by Formulator, another third party Zope Product (Atem de Carvalho & Monnerat, 2007, p. 340). The combination of the concepts provided by Zope, ZODB, CMF and Formulator allows through the web development for ERP5. The development tasks that can be accomplished through the web without touching one line of code on the file system span from editing small tailoring scripts to the creation of whole new bolt-on modules that solely live in the object database.

Zope and its CMF also provide an indexing service for Documents, called cataloging in Zope parlance. ERP5 improves on that with its ZSQLCatalog component which implements an object-to-relational mapping scheme to store the indexing attributes of each object in a relational database. Objects are stored in the ZODB, but search and retrieval are accomplished by fast relational databases using the Structured Query Language (SQL) which is a standard query language for relational data (Atem de Carvalho & Monnerat, 2007, p. 341; Smets-Solanes & Carvalho, 2003,

p. 40). ZSQLCatalog is currently used with the MySQL database, but ERP5 SQL queries are very simple and do not use any special database features to keep compatibility with other open source SQL databases (Smets, 2007). Reports can be either in python or SQL (Honoré, 2010a). The combination of an object data store and relational indexing allows ERP5 to deal with structured as well as unstructured data.

Apart from the mentioned core components around Zope technologies, ERP5 integrates a number of other open source software libraries and programs to accomplish specific tasks. Examples are Open Office for document conversion, reportlab for PDF forms and imagemagick for image transformations.

Generic features of the ERP5 core system such as activities, cataloging, common data structures and the implementation of the abstract core model are provided as ERP5 Zope Products (Honoré, Robin, & Smets, 2010) which together form the ERP5 Framework. This core is then specialized through Business Templates to provide applications in each business field of an ERP. Business Templates consist of Extensible Markup Language (XML) code which instantiates modules that provide the actual ERP functionalities (Smets, 2007, p. 10). They define one or multiple modules or provide extensions for modules defined in other business templates.

The ERP5 Base business template provides for example the Currencies, Persons and Organizations modules. The ERP5 CRM Business Template contains all modules related to Customer Relationship Management namely the Campaigns, Events, Meetings, Sale Opportunities and Support Requests modules (Rother, 2007, p. 34). Some Business Templates provide industry branch specific functionalities, like ERP5 Apparel or ERP5 Banking. Others are a specializations of existing Business Templates. In the public ERP5 repositories there are for example currently 13 Business Templates which apply the general accounting Business Template to the specific accounting standards of a particular country.

Business Templates are deployed as packages and can be imported into the object database of an ERP5 instance on demand. Atem de Carvalho and Monnerat (2007, pp. 347–351) show how new ERP functionalities can be implemented quickly with Business Templates that are developed on top of existing Business Templates reusing the abstract core model. Similar, the prototype developed during this thesis could be implemented in a relatively short time by reusing the ERP5 Workflow Business Template and core Product (see chapter 5: Prototypical Implementation, p. 49).

2.4.2 Unified Business Model

ERP5 defines and underlying abstract core model which is the base for representing all kinds of business processes. Figure 3 outlines this Unified Business Model (UBM) described by Smets-Solanes and Carvalho (2003, p. 42). The Name ERP5 is derived from the five abstract core classes *resource*, *node*, *movement*, *path* and *item*, that make up the UBM and help to consistently implement new or specialized components.

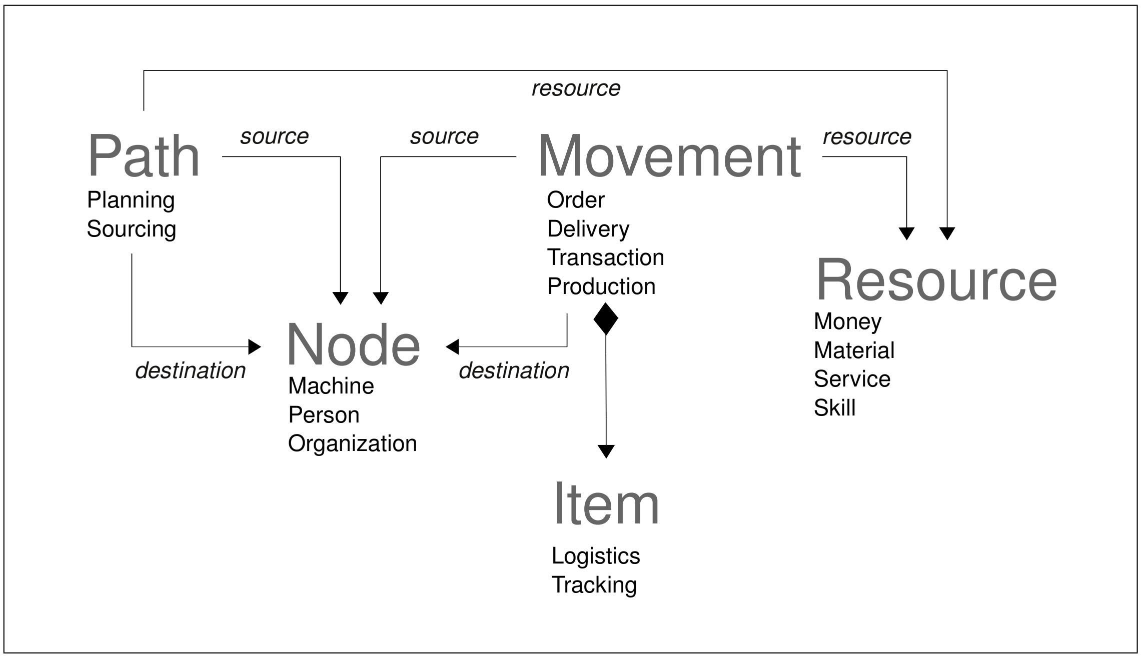


Figure 3: ERP5 Unified Business Model. The arrows symbolize relationships, the trapeze means composition. Adapted from Smets-Solanes (2002, p. 6) and Atem de Carvalho and Monnerat (2007, p. 342).

Resource is the base class for all abstract resources that are needed to realize a business process. Examples for resources in ERP5 are skills, that then are assigned to persons, currencies, services, products and components of products. Relations between resources are defined for example to describe the needed components for products in Bills of Material (BOMs) for PDM and MRP.

Node is a business entity that sends and receives resources. A node can be a physical entity, like a factory, that sends products and receives raw material, or an abstract entity, like a bank account that sends and receives money. Nodes have capacities which are either stock capacity or production capacity. Minimum and maximum amounts of resources that the node can contain or produce are defined in inequalities that the node must satisfy.

Movement describes the movement of a resource from a source node to a destination node. For example the shipping of a product from a supplier to a client is a movement that can be represented by a delivery line in a packing list. The payment following the delivery is another movement where money is sent from one bank account to another bank account.

Path defines the way and the conditions how a destination node receives a resource from a source node. As a trade condition it can define the standard price of a resource for a certain client or from a certain supplier. Paths also represent assignments of persons to projects for a period of time, so they can have a start and an end date.

Item is a physical instance of a resource. Items split an abstract movement of a resource between two nodes into movements of traceable items that can have a serial number and can define how they are being shipped.

Figure 3 outlines the relations between these abstract core classes and gives examples of their use. A movement contains multiple items and is related to a source node, to a destination node and to a resource that is moved between the two nodes. Similar, a path is related to a source and destination node and to the resource whose path attributes it defines.

The use of the five abstract classes for implementing business processes is best explained in the (simplified) example of a packing list generated in a sale trade process. A packing list represents a delivery and each delivery line is a movement. It contains the sender (source node), the recipient (destination node) and the product (resource) as well as a list of items, each with a serial number and a location for tracking as well as packaging information.

A packing list can be automatically generated from an order where each order line corresponds to a line in the packing list. In an order, source and destination nodes mean supplier and customer which might be different from sender and recipient in the delivery movement. The order can be based on a sale trade condition that is the path that defines the attributes like the default price per quantity of a product (resource) that is agreed between two organizations (nodes).

In each business process that reuses the core classes, they represent different business entities that share the same concepts but might have additional attributes and specialized behavior. Atem de Carvalho and Monnerat (2007, pp. 346–347) show how project management is implemented for ERP5 using the same concepts and core classes as the trade process.

Thanks to its technical architecture and abstract model, ERP5 gains some specific features and innovative ways to implement standard ERP functionalities. One ERP5 instance can be used across multiple organizations with multiple currencies and languages. Sites with weak Internet connection can run ERP5 by itself in the case of network failure and later be synchronized with the main ERP5 instance. The UBM provides *Simulation* functionality based on causality trees. Business rules transform one movement into another movement. A delivery movement for example generates an invoice movement, depending on the trade conditions between the supplier and the customer. Each invoice movement then generates a future payment movement. ERP5 supports *variations* of resources which can be defined by a collection of options such as color or size. This concept is extensively used in apparel industry and avoids having many different records for almost identical resources. Smets-Solanes and Carvalho (2003) describe the mentioned concepts and functionalities in more detail.

2.4.3 Meta Planning with Categories

Categories are a basic ERP5 principle which is subject to configuration in an ERP5 implementation and as such justifies a particular explanation. Categories help to classify business objects and to build hierarchies. Every business object in ERP5 can be associated to one or several categories. Categories can aggregate multiple nodes into a meta node or multiple resources into a meta resource. The *group* category for example is used to represent larger organizations which might be holdings for several smaller companies which again might have several subordinated departments. Meta nodes and meta resource can also be assembled using rules defined in predicates. A meta node “small retailers” could be defined using a predicate rule that depends on the business volume. Meta nodes and meta resources can act just like their non-meta counterparts. A supermarket supplier warehouse might categorize several products into a common product line category hierarchy “food / dairy products”. He could then sell this meta resource to the meta node “small retailers” the same way he would sell one single product to one retailer.

Categories are extensively used throughout ERP5. Examples are accounting (account type, financial section), human resources (function, skill), PDM (product line, quantity unit), document management (publication section) and CRM (role). Some categories are more or less the same between different ERP5 instances. An example is quantity unit which includes length, length/meter, length/centimeter, length/inches, weight, weight/kilogram, weight/gram, etc. Others, like industry *activity* have to be configured individually during an ERP5 implementation process to be usable. A more detailed description of selected categories, their use in ERP5, their configuration, and automation approaches for category configuration can be found in chapter 3.4 (Category Configuration, p. 26).

2.5 TioLive: Total Information Outsourcing

The aim of facilitating the implementation of ERP5 for SMEs lead to the development of TioLive, a free SaaS based on ERP5. Originally launched as ERP5 Express in September 2007, TioLive offers a preconfigured version of ERP5 especially aimed at SMEs (Nexedi SA, 2007). Within the One Student One ERP (OSOE) project it also provides TioLive instances to students and researchers of partner institutions including teaching material released under the Creative Commons License (Honoré, 2010b).

The name “TioLive” refers to Total Information Outsourcing (TIO), a term suggested by the Foundation for a Free Information Infrastructure [FFII] (2009). The TIO Libre initiative defines TIO as “a management approach which consists for a company to outsource all common information related operations to a collection of Web based online service providers” (TIO Libre Non Profit LLC, n.d.-a).

TIO Libre is a community of service providers that adhere to a series of rules to implement the TIO approach in a way that guarantees customers the same level of control and freedom as with the roll-out of free / open source software. TIO Libre is based on three principles: free access to source code, full access to ones own data and unrestricted competition (TIO Libre Non Profit LLC, n.d.-b). The objective of these principles is that a customer adopting such a service can at any time change the service provider or abandon the TIO approach by recovering all its data in native format, including the logs, from its original TIO service provider. TioLive is one of the companies adhering to the TIO Libre principles (TIO Libre Non Profit LLC, n.d.-c). With

TioLive Grid it allows users to install the TioLive appliances on their own servers and use the TioLive management interface to manage their private TioLive Cloud. The advantage of this way of adopting the TIO approach is that users have total control over sensitive business data (TioLive LLC, 2010).

TioLive not only provides preconfigured ERP5 instances, but also communication services that are integrated with ERP5's contact management, namely email, encrypted business chat and Voice over IP (VoIP) (Honoré, 2009). These services are based on open source software using open standards, mostly defined by the Internet Engineering Task Force (IETF): Email is provided by the Dovecot IMAP server, chat by ejabberd using XMPP/Jabber and VoIP by SIP Express Router (SER) using the SIP and RTP protocols (Smets, 2009).

TioLive includes selected ERP5 Business Templates which are pooled into preconfigured applications. The free version of TioLive currently includes Accounting, CRM, PDM, Trade and a Document Management System (DMS). Applications that require extensive configuration, namely MRP, Payroll, Project, Web and e-commerce are available on paid subscription (TioLive LLC, 2009).

2.6 TioLive's Configuration System

The TioLive infrastructure includes technology to facilitate basic configuration of a TioLive instance and the creation and installation of customized Business Templates. It consists of two main components, ERP5 Configurator and ERP5 Wizard. The Configurator is installed in an ERP5 instance on a configurator server. It includes Witch Tool, a generic utility that generates custom Business Templates from a data source. The general configuration steps for new TioLive instances are defined in a special configuration workflow. Each configuration step corresponds to a transition in the configuration workflow with an associated configuration script and a form for user input. The information entered by a TioLive customer in these configuration forms is the data source from which Witch Tool builds the Business Templates.

When a customer creates a new TioLive instance, a corresponding Business Configuration Document is added on the configurator server. On accessing a special configuration URL on the newly created TioLive instance the configuration forms are displayed to the user by the ERP5 Wizard Tool. The Wizard on the TioLive instance and the Witch on the configurator server talk to each other using the XML-RPC protocol: The Wizard asks the Witch for the next configuration form, displays the form to the user and then sends back the user's inputs to the Witch. After each configuration step a save point is added to the corresponding Business Configuration Document which contains the user's input for the current configuration step. This procedure assures that configuration can be suspended at any time and later resumed. At the end of the configuration process the Witch generates the customized Business Templates based on the configuration save points and sends them to the Wizard who then installs them on it's TioLive instance.

The prototypes presented in chapter 5 (Prototypical Implementation, p. 49) are meant for future integration into ERP5 Configurator. The TioLive configuration wizard would then display decision trees and questionnaires in addition to the current hand-made configuration forms. The current configuration scripts would be supplemented by automatic configuration logic based on expert knowledge and data mining.

3 ERP Package Tailoring

Following the previous introduction to ERP5, TioLive and its configuration system, this chapter discusses the applicability of different types of ERP package tailoring to automation. Furthermore it presents selected ERP5 configuration options in an exemplary configuration case.

3.1 Tailoring Types Best Suited for Automation

Adopting an ERP package requires its adaption to the specific business needs of an organization. ERP systems are often viewed as *off-the-shelf* software, which means that they are usually implemented by setting parameters in the package to adapt their functionality to business requirements. This form of implementation, called *configuration*, is usually distinguished to *modification* which refers to changing the source code of a package. Modification is considered typical for *custom-built* software (Brehm et al., 2001, p. 1).

Brehm et al. (2001, p. 2) argue that ERP systems do not fit into this traditional distinction. They use the term *tailoring* to refer to both, configuration and modification as well as many options in between. They suggest a typology of nine different ERP tailoring types shown in Table 1 (Typology of ERP tailoring types, p. 16). The order in which the tailoring types are presented in the table is roughly derived from the "impact" they have on the ERP system as well as on the ERP adopter, beginning with "lighter" tailoring types at the top of the table to "heavier" tailoring types at the bottom. For the ERP system, impact means how severely it is being changed if a tailoring option is applied. For the ERP adopter, impact means how much effect is required to employ a tailoring type (Brehm et al., 2001, p. 5).

The impact of tailoring on the ERP system also affects automation. The heavier the ERP system is changed through tailoring, the more complicated is the required automation logic to facilitate the tailoring. *Configuration* is the tailoring type with the lowest impact. The possible values for each configuration parameter are bounded by the value range of the configuration parameter. They can be further narrowed by defining a set of configuration cases that the automation method should support. This set could contain:

- all theoretical possible configuration cases,
- all realized configurations in the past plus a predefined set of possible values or
- a set of viable values defined by a function.

Thus, the solution space of automated configuration is bounded and configuration is predictable.

Brehm et al. (2001, p. 5) place bolt-ons at the low-impact end of their typology. They remark that the impact of bolt-ons is debatable because their quality depends on the communication between ERP vendor and bolt-on developer. They further argue that the risk of a release-lag between

Tailoring Type	Description	Layer
Configuration	Setting of parameters to choose between different executions of processes and functions	All layers
Bolt-ons	Implementation of third-party package designed to work with ERP system and provide industry-specific functionality	All layers
Screen masks	Creating new screen masks for data in- and output	Application and/ or database layer
Extended reporting	Programming of extended data output and reporting options	Communication layer
Workflow programming	Creating of non-standard workflows	Application and/ or database layer
User exits	Programming of additional software code in an open interface	Application and/ or database layer
ERP programming	Programming of additional applications, without changing the source code (in vendor's computer language)	All layers
Interface development	Programming of interfaces to legacy systems or third-party products	Application and/ or database layer
Package code modification	Changing the source-codes ranging from small changes to change whole modules	Can involve all layers

Table 1: Typology of ERP tailoring types adapted from Brehm et al. (2001, p. 4).

the ERP system and bolt-on version can be an issue when updating the ERP system. Both considerations also apply for the suitability of including bolt-ons in an automated tailoring process. Choosing, installing and configuring bolt-ons can be suitable for automation, if ERP vendor and bolt-on developer collaborate on the automation process. Especially open source business models favor bolt-on automation. If the source code of the ERP system and the bolt-on are available in public Version Control Systems (VCSs), then the integration and automatic configuration of bolt-ons can be automatically tested for each version of the ERP system and the bolt-on.

For heavier tailoring types, automation logic would be more complex. Source code often has to be generated automatically. For *screen masks*, *extended reporting* and *workflow programming*, a combination of automation with easy to use graphical design tools could be a solution. For *User exits*, *ERP programming* and *interface development*, an automation system could gather the required information from the user through questionnaires and generate a rough code structure that would be the base for a final implementation by a human consultant. Whether the higher tailoring types are suitable for automation also depends on how generic functionalities are implemented in the ERP system and how easy it is to reuse existing data models and functions as building blocks for new functionalities.

In package code modification, the automation would have to “understand” the whole ERP system to be able to do modifications and calculate their impact. Also, the automation system itself would have to be adapted on every update of the ERP system. The automatic generated code would then have to be regenerated automatically to reflect the changes.

The impact of tailoring on the ERP system is not only affected by the type of tailoring, but also by *how extensively* a tailoring type is used (Brehm et al., 2001, p. 5). This factor also influences the suitability of tailoring for automation. Using the configuration type of tailoring more extensively means for automation that more configuration options are automated and that for each configuration option, more configuration cases are considered by the automation system. A system, that automates more extensive configuration, has to ask more questions to the ERP adopter in the configuration process and consists of more complicated automation logic, for example larger and thicker decision trees. The influence of extensiveness to automation can be seen in comparing the draft version and the improved version of the site decision tree (chapter 5: Prototypical Implementation, p. 49). In its draft version, the site decision tree aimed to support a higher number of the theoretical possible configuration cases with a deep site hierarchy. This resulted in a complicated decision tree consisting in too many and too difficult questions. The improved version of the decision tree achieves with reduced complexity faster and easier configuration for many standard configuration cases relevant to SMEs at the cost of leaving apart some edge cases covered by the draft version.

The mentioned considerations indicate that the impact of tailoring on the ERP system might be a viable indicator for the suitability of tailoring for automation. Therefore, the thesis is based on the hypothesis:

The lower the impact of tailoring on the ERP system, the more likely it is that the tailoring is suitable for automation.

Following this hypothesis, configuration is the tailoring type which is most likely suitable for automation, depending on how extensively it is used. Therefore, the automation approaches and prototypes presented in the following chapters concentrate on automating the configuration type of tailoring. Automating other tailoring types like *screen mask* generation, *extended reporting*, and *workflow programming* is topic for future research. Once automation has been implemented for multiple tailoring types with different degrees of impact, the hypothesis could be tested by analyzing the effectiveness of the automation implementations and the effort necessary to implement them.

Brehm et al. (2001, p. 7) argue that tailoring increases the degree of fit between the features and functions of an ERP package and the business processes of a particular organization. They hypothesize that “the greater the impact of tailoring on the ERP..., the more likely it is that...the system will meet the needs of the business.”

Configuration is the tailoring type with the lowest impact in Brehm et al.’s typology. From an ERP adopter’s point of view that means that if the gap between the ERP package functionality and the business needs is too big to be filled by configuration, tailoring types with greater impact are required. From an ERP package design point of view however, an alternative to automate tailoring types with a higher impact can be considered: designing the ERP package to be more generic and to offer wider configuration choices.

Following this alternative, automating consists of two parts: Automating the configuration of an ERP package and enhancing the ERP package in a way that more tailoring tasks that currently require tailoring types with a higher impact can be achieved solely by configuration. This is why the investigations conducted during this thesis concentrate on the automation of the configuration type of ERP package tailoring.

The next chapter presents a categorization of ERP5 tailoring options into the tailoring typology. Further, it explains how a broad range of implementation tasks in ERP5 can be conducted by employing the configuration type of tailoring.

3.2 Tailoring Options in ERP5

Most of ERP5's tailoring options don't map unambiguously to Brehm et al.'s tailoring typology, however a rough categorization is presented in Table 2 (Categorization of ERP5 tailoring options, p. 19). Similar to Brehm et al.'s typology which refers to the general three-layer model of application systems, the last column in table 2 refers to the three layers in ERP5, where tailoring can take place: Business Templates, Property Sheets and Zope Products.

Business Templates assemble applications from configuration parameters, forms, views, reports, workflows, document types based on ERP5 core classes and Property Sheets, modules, custom scripts or actual documents based on ERP5 document types (see chapter 2.4.1: Technical Components, p. 7). All these are objects in the ZODB. Most tailoring is realized in this object space through-the-web. The first configuration step of a new ERP5 Instance is the installation of the required business templates. Basic automation for this procedure is already implemented in ERP5 Configurator (see chapter 2.6: TioLive's Configuration System, p. 13). Then, tailoring is conducted by setting attribute values of existing objects, copying and modifying objects or creating new objects. These objects can be packaged as a new custom business template containing the results of all tailoring at this level.

Referring to the general three-layer model of application systems, the implementation of the communication layer is completely contained in Business Templates. The “PortalSkins” tool which is part of the CMF is used to manage forms, corresponding action scripts and page template views in layers, called “skins”, which allow to customize all user interface related objects without touching the originally installed objects. Interface methods to communicate with other application systems over XML-RPC are also defined in PortalSkins. Business Templates are also involved on the application-layer as they contain workflows, reports and the actual ERP5 modules and document types which are assembled through-the-web based on the core classes and Property Sheets.

Property Sheets define the ERP5 data model. Each property sheet can be reused by any core

Tailoring Type	ERP5 Tailoring Options	ERP5 Layer
Configuration	Choosing modules and workflows, defining site preferences, categories, business processes and security	Business Templates
Bolt-ons	Installing third-party Business Templates; adding Property Sheets / Zope Products, if data model extensions / auxiliary core functionalities are required	Business Templates, Property Sheets and Zope Products
Screen masks	Creating form views, fast input forms and Page Templates in custom skins	Business Templates
Extended reporting	Designing search forms and create SQL- or Python reports with ERP5 Report Wizard	Business Templates
Workflow programming	Creating custom workflows and implement associated actions and worklists	Business Templates
User exits	Creating new modules, document types, actions, forms, jumps and interactions based on existing types	Business Templates and Property Sheets
ERP programming	Creating Zope Products to provide core extensions or to integrate external libraries	All layers
Interface development	Designing XML Import- and Export conduits; Creating python scripts for invocation through XML-RPC	Business Templates and Property Sheets
Package code modification	Modify core ERP5 Business Templates, Zope Products or standard Property Sheets - not meant to be necessary by ERP5 design philosophy	All layers

Table 2: Categorization of ERP5 tailoring options.

class as well as by document types defined in Business Templates. Although arbitrary attributes can be set and accessed in the ZODB for rapid prototyping, automatic indexing and dynamically generated accessors are only available for attributes defined in Property Sheets. However, as Document types, forms and workflows can easily use attributes already defined in existing Property Sheets and as ERP5's standard attributes are very generic, the need for new properties is very rare (Gorny, Nowak, & Perrin, 2008). Whole new modules can often be designed by assembling document types out of multiple standard Property Sheets.

Zope Products are the place where ERP5 core components are defined, including the classes of the UBM (see chapter 2.4.1: Technical Components, p. 7). As they contain core application logic, they are comparable to the application layer of the general three-layer model of application systems. ERP programming can involve the development of Zope Products, if extensions to the ERP5 core model are required or external libraries should be included. Bolt-ons might provide additional Zope Products to extend the ERP5 core model for industry-specific requirements. Zope Product development would also happen in case of package code modification, though theoretically it should not be necessary in an ERP5 implementation process (see p. 21).

The ERP5 tailoring options have been allocated in Table 2 (Categorization of ERP5 tailoring options, p. 19) to tailoring types according to their maximum possible impact weight. In practice, the impact of most tailoring options is lower than the usual impact of the tailoring type to which they have been assigned.

Configuration is the most commonly used tailoring type in ERP5. Many tailoring options that involve high-impact tailoring types in complex cases can be accomplished solely by configuration in simple cases. Definition of site preferences is the tailoring option that fits best to the traditional understanding of configuration as it consists of instance-wide configuration parameters that alter the behavior of ERP5 functionalities. TioLive's configuration systems already automates some configuration-type tailoring options like the choice of locale dependent accounting business templates, configuring default site preferences and generation of initial documents, for example the adopter's organization and its employees.

Bolt-ons have a low impact in ERP5 if they consist solely of Business Templates as they can be installed automatically and assure version compatibility. The impact will be higher if bolt-ons include core extensions in form of extra Zope Products that subclass ERP5 core classes to add industry-specific functionalities.

Screen masks are assembled in ERP5 out of configurable form field objects. This type of tailoring mainly consist in creating "fast input forms" to optimize data input for critical business processes (Smets, n.d., p. 36). The impact of ERP5 fast input implementation is higher, if complex user interfaces are designed for special purposes like point of sales.

Extended reporting is conducted in ERP5 to extend standard PDF rendering forms with statistics such as inventory, average price or with custom visual design . ERP5 Report Wizard helps to generate reports with SQL- or Python reports (Smets, n.d., p. 37). Different reporting engines are available using direct PDF generation with reportlab or OpenOffice for generating reports in different office formats.

Workflow programming will be required, if ERP5's standard workflows are not sufficient to fulfill the adopter's needs. In simple cases, only configuration of standard workflows is necessary, for example:

- mapping workflow states to tasks in the company,
- changing the rules, that control if a transition gets activated or
- adding intermediate workflow states to support more complex decision processes of an adopting organization (see Smets, n.d., p. 35).

If new actions are programmed or the implementation of existing actions is changed, then programming of python scripts will be involved and impact of this tailoring type will be possibly higher.

User exits and *ERP programming* cannot be clearly separated in ERP5. New modules with new document types can be designed in ERP5 in a way that is closer to configuration than to programming. Since forms and document types are just objects in the ZODB, the impact of creating new modules or changing existing modules merely depends on how many existing objects, actions or Property Sheets can be reused in the process. This is additionally supported by techniques like "Proxy Fields" that make sure that custom form fields based on other existing form fields adapt automatically to changes in new versions of ERP5 (see Courteaud, 2009, p. 8).

The sum of user exits provided by ERP5's through-the-web programming system enables ERP programming with an impact similar to the impact of configuration in simple cases. In complicated cases, the impact of ERP programming in ERP5 is similar to the impact of user exits. The particular impact also depends on the level of reuse. Therefore, Table 2 (Categorization of ERP5 tailoring options, p. 19) allocates this kind of configuration options to:

- to user exits, if they only involve the Business Template layer or the Property Sheets layer,
- to ERP programming, if they involve creating new Zope Products to provide core extensions or to integrate external libraries.

Package code modification should theoretically not happen in an ERP5 implementation process. ERP5's design philosophy strives to make the package code general enough to accomplish adaption to different business needs by employing lighter tailoring types. In cases where the core package code is still not general enough, the open source nature of ERP5 favors the improvement of package code in the public source code repositories over custom changes that only serve one adopter and aggravate package updates. In practice however, due to time constraints in implementation projects, package code modification might be first employed for a customer before they are integrated back into the main ERP5 branch.

Thanks to its abstract model, ERP5 is a very generic application system, thus many ERP5 tailoring tasks can be accomplished solely by configuration and still have great effect on how ERP5's functionalities behave. This isn't fully reflected in table 2, therefore Table 3 (ERP5 implementation tasks., p. 22) categorizes ERP5 tailoring by implementation tasks. These are described in the ERP5 implementation process (Smets, n.d.). The implementation process contains analysis, implementation and test phases. Analysis is based on interviews and document research. Its purpose is to discover resource flows and decision flows in a company. It also aims to identify the demand for implementation of custom document types. The procedure is aligned to ERP5's document-centric approach to implement business processes.

The implementation process is supported by ERP5 through a series of tools for requirements, analysis, design and implementation as well as through general process related tools (Atem de Carvalho & Monnerat, 2008). Table 3 (ERP5 implementation tasks., p. 22) presents a simplified version of the implementation process. It focuses on the implementation tasks that consist in actual tailoring and omits the analysis and testing tasks. The purpose of the figure is to show which tailoring types can be involved in each implementation task.

Group definition and *Categories* can be implemented purely by configuration and are therefore

Implementation Task	Tailoring Types
Group definition	Configuration
Categories	Configuration
Core extensions	bolt-on or user exits and/ or ERP programming and/ or package code modification
Decision implementation	Configuration and/ or workflow programming
Document implementation	Configuration and/ or screen masks and/ or user exits
Search & report implementation	Extended reporting
Integration	Configuration and/ or workflow programming and/ or extended reporting

Table 3: ERP5 implementation tasks.

considered suitable for automation. Still, they have a great effect to the behavior of ERP5. The automation approaches presented later in this thesis are applied to automate category configuration. Group definition is related to category configuration, as users can be assigned to security groups dynamically based on multiple categories. Therefore, group definition and category configuration deserve a more detailed description in the next two chapters.

The implementation process of ERP5 contains an analysis task, called abstraction test (Smets, n.d., p. 28). Its purpose is to find out if ERP5's core model is able to represent the document flows in the adopting business. If some document flows cannot be modeled, *core extensions* will be required. In some cases, core extensions already exist in form of industry-specific bolt-ons, for example "ERP5Banking", which consists of an additional Zope Product and specialized business templates. Otherwise, core extension development is a tailoring task with high impact, that usually consists in developing additional Zope Products and complementing business templates.

Decision implementation, *document implementation*, *search and report implementation* and *integration* are part of the module implementation phase in the ERP5 implementation process (Smets, n.d., p. 34). Individually, these tasks can be applied to alter the behavior of existing modules according to the needs of the adopting organization. Implemented together, they form a new module.

Decision implementation consists in configuring ERP5 workflows to fit the requirements of the business (Smets, n.d., p. 35). Workflows can be created using ERP5's workflow engine in a configurable way. Decision implementation includes two sub-tasks:

- worklist implementation and
- action implementation.

Worklist implementation consists in defining how states in a workflow map to tasks in an organization. It can be accomplished by configuration. Action implementation consists in defining transitions that are meant to be executed by users. It might also include the development guards which control abstract transitions that are triggered automatically. This goes beyond configuration and can be assigned to the workflow programming tailoring type.

Document Implementation consists in altering existing document types or creating new document types. In most cases, the ERP5 core data model can be reused by configuring new document types to use viable existing property sheets. Also functionality can be reused by mapping document types to one of the ERP5 core concepts described in chapter 2.4.2 (Unified Business Model, p. 9) and to existing document classes already implemented in ERP5 (Smets, n.d., p. 36). Although this task can be accomplished purely by configuration, it requires deep knowledge of ERP5. Depending on the adopter's needs, document implementation might also include:

- type implementation and
- fast input implementation.

Type Implementation consists in adding or customizing forms, views and menu actions. It can furthermore include the creation of new property sheets, if some needed document fields cannot be mapped to existing property sheets. Thus, type implementation can be assigned to the screen mask and user exits tailoring types. Fast input implementation consists in implementing specialized forms with massive lists of fields to optimize data input for critical business processes.

Search and Report Implementation maps well to the extended reporting tailoring type whose implementation in ERP5 has been described above (see p. 20). Search forms and reports that are created during module implementation concentrate on data defined in the respective module.

Integration consists in turning the independent modules that were implemented during module implementation into an integrated application (Smets, n.d., p. 40). It contains

- jump Implementation,
- interaction implementation and
- report implementation.

Jump implementations consists in adding actions that help users to quickly jump from one document to related documents for example from a sale order to a related packing list to a related invoice.

Interaction implementation consists in creating special interaction workflows whose transitions are triggered automatically to implement behavioral relations between independent modules. For example, if the invoicing of a customer should also change the state of a trade condition related to that customer, it should be implemented through an interaction workflow assigned to the accounting module. Then, an interaction is created inside the interaction workflow and configured to be triggered by the “Journalize Transaction” action of the accounting workflow. Finally, a Python script is added, that triggers the “Invalidate” action of the validation workflow of the sale trade condition related to the customer who has bee invoiced.

Jump implementation and interaction implementation can be assigned to the configuration and workflow programming tailoring types. Their impact merely depends on how complicated the trigger rules are. Report implementation can also be part of the integration tasks. It consists of creating search forms and reports that combine data from different modules, and therefore cannot be implemented as part of module implementation.

Table 3 (ERP5 implementation tasks., p. 22) shows that many tasks in the ERP5 implementation process include configuration type of tailoring. The above description of these implementation tasks shows that some of them can be accomplished solely by configuration in standard situations that don’t require intensive tailoring. This indicates for the case of ERP5, that not only pure configuration tasks are suitable for automation, but also the tasks, that would be usually associated with higher impact tailoring. Automating these implementation tasks is a promising topic for future research.

3.3 Group Definition

Security in ERP5 is based on Zope’s role-based security system. Each document has permissions, for example to create, view, modify or delete the document or to view its version history. The Zope security system allows to define roles and associate them to users. The permissions of each document can be assigned to roles. For example, a user x might be assigned the role “Accounting Manager” and the “view” permission of the account “Refundable VAT” might be assigned to the roles “Principal Cashier” and “Accounting Manager”. In this setup, user x has the permission to view the “Refundable VAT” account. Zope’s “local roles” concept enables a user to have different roles on different documents. This makes complex security configurations possible, but the number of roles can become very large in ERP applications and makes it hard to manage security.

To make security more manageable, ERP5 uses the concept of group indirection to reduce the number of roles (Perrin & Smets, 2010, p. 10). For security configuration, ERP5 uses a minimal number of five generic local roles:

- *Author*, the document creator;
- *Assignor*, who assigns a document to an
- *Assignee*;

- *Auditor*, who has complete access to document contents and
- *Associate*, a participant in document processing.

This helps in managing security efficiently, even if the number of users and groups grows to very large numbers. Hence, instead of assigning user x a role directly, he is member of the group “Accounting Manager” which itself is associated to the role “Assignor”.

ERP5 improves on the concept of security groups by assigning users to groups automatically based on categories. Each category can have a security codification. Group names are constructed out of the codifications of the categories they depend on. In a small mutual savings bank, security might be defined based on the categories function, group and site, so that for example some accounts can only be viewed by accounting managers of the bank’s private banking branches in London. In this example, there could be a category “function/accounting/manager” with codification “ACM”, a category “group/british_mutual/private_banking” with codification “BMP” and a category “site/london” codified as “LND”.

To implement this example, the mentioned categories are defined in category configuration. During group definition, a new role information document is added for the account document type. The “role” attribute is set to “Assignor” and the three categories are entered in the “category” field for local role name generation. A condition can be set to apply this role only to certain accounts. This configuration results in a local role “Assignor” that is associated to a group named “ACM_BMP_LND” for all accounts that fulfill the condition. A user’s membership to this group is calculated dynamically based on the user’s active assignments.

In chapter 2.4.2 (Unified Business Model, p. 9) it has been mentioned that a person in ERP5 is derived form the node core class and that paths can be assigned to nodes, which in this case is a career step called for example “Branch Accounting Manager, Private Banking” assigned to user x. The career step document has fields for function, group and site. If these fields are set to above mentioned categories - or subcategories of them, for example “site/london/west_end” - then user x is automatically member of group “ACM_BMP_LND”. As such, he has the local role “Assignor” on the configured accounts. Although the group membership is calculated automatically, this kind of security configuration is termed *static security*, because the categories that are used to build the security group are configured statically.

Dynamic categories will be applied in security configuration, if the security group should be generated dynamically from a category of a related document (Perrin & Smets, 2010, p. 13). In the example above, the account transactions might be entered by an accountant clerk y, and the account manager x is responsible for validating these transactions. User x should only have the permission to do this, if he has the function “Accounting Manager” and belongs to the same service and the same branch as the clerk. The security group name is generated out of group (which denotes the service), site (which denotes the branch) and function, like in the previous example, but this time the group and site categories are dynamically fetched from the person document of clerk y. To implement this behavior, a Python script is created that returns the values “group” and “site” of the person that created the account. The role information of the account document type is altered by defining “group” as a dynamic category and entering the name of the Python script. On accessing an account, the script is executed in the context of the account document and thereby has access to the attributes holding the group and site categories assigned to the account creator to dynamically create the security group.

Static group definition only involves configuration and is therefore considered being suitable for automation. This is a task for future research as this thesis concentrates on automation of category configuration. Dynamic security group generation is slightly more complicated because of the base category script. This script is usually quite simple as it only gets the category set to a related document. Thus, automatic generation of this script might be possible. Another approach would be to provide a set of standard base category scripts and then let the automatic system choose the right scripts.

3.4 Category Configuration

One of the configuration options that highly influences the behavior of ERP5 is category configuration. Categories help to classify business objects and to build hierarchies. They define not only the structure of the company, but also the company’s view of the world in a taxonomy.

3.4.1 Creation of Categories in ERP5

Categories can be directly added in ERP5’s PortalCategories tool. Categories belong to a “base category” which is the root of a category tree. For example “region” is the base category of the category “region/europe/france”. Technically, a base category can be described as a configuration parameter which accepts a hierarchy of categories as configuration value. To facilitate category configuration during ERP5 implementation, categories can be defined in a spreadsheet that is later imported into ERP5.

The compact disc attached to this thesis includes a sample spreadsheet (Aurora-Configuration.ods) containing the category configuration of the configuration case described in chapter 3.5 (Configuration Case, p. 33). Each sheet in the file belongs to one base category, each row defines a category. The columns A to I denote the *path* of the category in the hierarchy, *ID* is the identification of the category in ERP5. *Reference* is a unique name of the category in an ERP5 instance, which is necessary, if the category should be accessed without its path. *Codification* is used to construct the names of automatically generated security groups based on the category (see chapter 3.3: Group Definition, p. 24). *Title* is the name, as it is displayed in ERP5 forms, *short title* is used in situations where the user interface requires a short string. Description gives a more detailed explanation of the category.

3.4.2 Description of Selected Categories

Some of the categories which have to be configured individually during ERP5 implementation are presented below. Only the essential categories are described and only those that are required for ERP5 functionalities offered by TioLive. The information about these categories was gathered through analysis of previous ERP5 implementation projects, expert interviews, questionnaires and ERP5 consulting documentation (see chapter 4.1: Automation Procedure and Information Sources, p. 38).

“Site” is the category which is described in most detail because its configuration is later automated with a decision tree. The design of the decision tree required very detailed information about the site category and its configuration, A detailed questionnaire and several private conversations with staff of Nexedi SA were conducted to gather the required information about the site category. Expert questionnaires were also used for the group, role and region categories.

Site

Sites defines the physical structure of an organization. The category is structured as a tree of sites with children sites. Examples of site categories are “site / france”, “site / france / noyon”, “site / france / paris”, ”site / spain”, ”site / spain / barcelona”. The level of detail can differ from one site to another. For example, a small retail store site may be known only by its city or region name. A factory might be defined in six hierarchy levels, from the regional name to the identifier of a storage cell in one of multiple warehouses of the factory (Nexedi SA, 2009).

Site is used to define the physical location of organizations and persons. Persons can be temporary assigned to different sites through assignments. For example an employee of a consulting company might work at the site of a client during an implementation project.

Although names of countries or regions might be used as site categories, site does not define regional structure from a political or sales area point of view. That is the purpose of the region category later described in this chapter. Entities that are chosen for site categories should be physical and as stable as possible, for example continents, cities or buildings (Jean-Paul Smets, Nexedi SA, personal communication, March 23, 2010).

Site can also define a storage hierarchy. A storage node, which acts as source or destination for a movement is related to a site. Depending on the customer’s needs the actual storage node could be an organization related to a high level site (city name) or a storage cell (e.g. compartment) related to a low level in the site hierarchy (city / area / warehouse / floor / rack). To better support the latter case, a storage module will be added to ERP5 in the future, in order to create one level of indirection between storage nodes and the site category. The lowest level in the site hierarchy is just above the actual node respectively the storage object. The granularity of the storage hierarchy depends on the wishes of the adopting organization. It is possible to define a site category for each case in drawer in an optical store to record the exact storage of each glasses. But it might be more practical to just set the store as storage node in ERP5 and use labels on the drawers and a corresponding field in the item document to denote the exact storage position of an item (Questionnaire replied by Jean-Paul Smets, Nexedi SA, May 1, 2010, see appendix A).

Since site is one of the categories which are referenced by persons, it can be used for security group generation (see chapter 3.3: Group Definition, p. 24). If this functionality should be implemented, security codes must be defined for site categories at the security-relevant levels. For example storage cells of a warehouse might not be relevant for security. However, it can be useful to differentiate security permissions for each warehouse of the same factory. A warehouse clerk of warehouse A may be prevented from entering inventories for warehouse B (Nexedi SA, 2009).

Five previous implementation projects were reviewed for their site configuration. They are good examples for different configuration needs of different types of companies.

The SANEF project didn't need any site configuration. The company has only one office that is relevant for the ERP5 implementation. Therefore only the default site category "main" was used.

The NXD project is a good example for a typical site configuration for SMEs. Six different sites are defined by a flat list of city names where the company's offices are located. Although sites are distributed over several countries, a hierarchic configuration of sites doesn't make sense because of the low number of sites.

MEDICENTRE is a configuration project for a hospital. It consists of only one main site at the upper hierarchy level. This main sites is divided into 18 subcategories that refer to the different buildings and wings that are located at the hospital site.

The AUCKM project is an example for a larger inter-governmental organization with many sites in different countries and continents. There are between 2 and 22 sites per continent. Therefore, sites are structured in two levels, first by continent, then by city name - a further structuring per country would not make sense in this case, as there is only one site per country.

The most extensive use of site configuration of the reviewed implementation projects has been employed in the BAOBAB project, an ERP5 implementation for a central bank. Several hundreds of site categories are configured in a five level hierarchy. The hierarchy levels are type of building (two levels), city, floor and area.

The review of previous implementation projects showed that sometimes sites are named similar to other categories, for example group, region or function. Larger organizations might have many sites, so they want to structure them by regions. Some of these regions can overlap with sales area regions which are configured in the regions category. This overlap can lead to confusion, if for example Japan is considered as part of Asia in site configuration and as a continent of its own from a sales area point of view (see paragraph: Region, p. 30). Overlap with group or function can happen because one often refers to a site name by the department or business unit that is located at the site. For example, in the BAOBAB project, the sites “agence / dakar” and “siege / dacar” refer to two different buildings in the city Dakar. “Agence” is located at one building and “Siege” at the other building, so the organizational units were mixed into the physical site structure. An automated configuration system has to take possible confusion between categories into account. This can be done in giving precise definitions during the questioning process and explaining the purpose of the questions.

Group

Group defines the juridical structure of an organization with its subsidiaries and business units. Group categories can also define departments / sub-departments or divisions / sub-divisions, even if they are part of the same “juridical group”. The category is structured by the concept of subordination. The purpose of the group structure is to describe how responsibilities and power are delegated and subordinated in a company. Groups are assigned directly to an organization. Persons acquire groups indirectly from the organization that is associated to the career or assignment of the person. The group category structure is used for three main functionalities in ERP5:

- security,
- human resource management and
- analytical accounting.

Like the site category, group can be used for security group generation (see chapter 3.3: Group Definition, p. 24). If this functionality should be implemented, security codes must be defined for group categories at the security-relevant levels.

Similar to security configuration, group together with site and function defines positions in a company and is therefore used for human resource management. A person in a higher level group hierarchy usually has more decision power and responsibility than a person whose career is assigned to a lower level in the group hierarchy (Questionnaire replied by Jean-Paul Smets, Nexedi SA, April 18, 2010, see appendix A).

The structure of the group category shows how responsibilities are delegated across the divisions or business units of a company. Therefore, group is also used for analytical accounting to analyse profit and loss per group on different hierarchy levels. In this case, groups are defined as profit centers.

Group also defines ones own vision of the structure of third party business entities (clients, suppliers, partners). For example, if an adopting company does business with a group of companies named “partner_group”, it might want to differentiate its two entities “infoservice” and “cleaning”, especially if only infoservice employees are its partners and if they are allowed to access its own ERP5 system within a specific partnership contract which grants them access to some documents

(Nexedi SA, 2009).

Role

Role defines a categorization of all persons and organizations that are stakeholder of the company. Examples are staff, clients, potential clients, suppliers, or medias. Clients might be further divided into direct clients and distributors, medias may have subscribers and associations have members (Nexedi SA, 2009). The purpose of the category is to unify the contact databases in ERP5. Instead of having a client database, a staff database etc., there is only on database, with consistent information. This way, the same person or organization can have multiple roles. For example a vendor in an optical store is an employee, but at the same time he can be a client with his own patient history. Roles can also be used to manage potential clients, like sales leads and sales prospects (Questionnaire replied by Jean-Paul Smets, Nexedi SA, April 18, 2010, see appendix A).

Function

The function category describes the functional structure of the adopting organization and of organizations that it is doing business with. Functions of organizations are the nodes in the function tree, for example “factory”, “warehouse”, or “factory / warehouse”. Functions of persons are represented by the leaves, for example “factory / manager” or “lab / director”. ERP5 distinguishes function and grade. For example, a person whose grade is director of research may be assigned the function of factory manager. Grades can be compared to army grades, for example “general” or “commander”. Functions are the assigned missions, for example “researcher” or “spy” (Nexedi SA, 2009).

Grade

Grade describes the position in an organization from a honorific point of view. Salary is also usually based on grade, rather than on function. Though in ERP5, salary levels can be defined in its own category. A typical example of grades are “general”, “commander” in the army. Some army generals are sometimes assigned to research management functions. Their grade is still general but their function is “director of R&D center”. There is no direct relation between grade and function. Grades differ from function in its meaning Function describes actual operational position while grade describes a honorific position (Nexedi SA, 2009).

Region

Region defines sales areas from a geographic-political point of view. The category is used to record the location of Organizations and Persons and to generate reports where sales and clients are clustered per sales areas (Questionnaire replied by Jean-Paul Smets, Nexedi SA, April 18, 2010, see appendix A). ERP5 defines some standard regions. However each adopting company should define its own view on regions to fulfill their specif needs and to cover sales. These are concepts such as EMEA (Europe, Midldle East, Asia). Moreover, the choice of a list of regions includes both commercial and political consequences. Taiwan, for example can be defined as a country or not, depending on the political or commercial point of view. Another example is Japan, which most inhabitants consider not to be part of Asia. Therefore, doing business in Asia often requires a special treatment for Japan, which is considered as a continent on its own, whereas China influenced countries are put in a single group.

Depending on the location of the ERP5 adopter, different regions are defined with different levels of detail. A European business might not care about states in the US. However, a US business needs to keep track of its clients based on the state information. US states should therefore appear in the region category for a US business. German business might define only one category for US, but divide Germany into several regions based on the Bundesländer. A local business may require more precised region categories, for example different areas in the same city (Nexedi SA, 2009).

Skill

Skills are assigned to persons and are part of the human resource management functionalities. They can also be assigned to people outside the adopting organization to record skills that for example consultants provide to the organization (Nexedi SA, 2009). Some skills can be defined universally, like languages or drivers licenses. Others are organization-specific, like the ability to control a specific machine.

Nationality

Nationalities are defined more universally than regions. Therefore, the standard categories can be used in most cases. Exceptions can be again nationalities like Taiwanese, where there is no universal consensus, if Taiwan is a nation on its own. A further level in the hierarchy can be defined in the case of groups of countries, for example the European Union (EU) or the Economic Community of West African States (ECOWAS). Their inhabitants carry a passport with a dual mention: the group of countries and the country itself (Nexedi SA, 2009).

Activity

Activities describe a company's view on all economic activities that are relevant for the company. There are branch-specific and universal standards for activities, for example the classification by United Nations Statistics Division (2008). However, experience of previous ERP5 implementation projects have shown that often the number of standard activities is either too high or the categories are not detailed enough for the specific requirements of a business (Interview with Jean-Paul Smets, Nexedi SA, November 2, 2009, see appendix A). Therefore, specific activities are usually defined for each ERP5 adopter. Activities are different from function. They usually relate to a classification of third parties based on the nature of their industry, for example "banking", "IT", "automotive". Function, on the other hand, is independent from the activity. Both, an IT and an automotive company can have an entity which has the function of a warehouse (Nexedi SA, 2009).

Publication Section

Publication Section is used to categorize documents per document type for retrieval in ERP5's DMS. A document type is for example a contract, a status report, a letter, etc. The different document types which are involved in your daily business can be stored in the DMS, either autonomously, for example a letter in form of an OpenOffice Writer document. Or it is stored in relation with a business document, for example a specification document related to a sale order, or an error report attached to a support ticket. Many ERP5 business document types support the assignment of documents, the publication section category helps to structure these documents (Nexedi SA, 2009).

Product Line

Product lines are used for both purchases and sales. They are useful to create a catalog of products (sold or purchased) and to structure a large database of products and services by families (Nexedi SA, 2009). Product line is also used for CRM to cluster customers and sales prospects by product interests.

3.4.3 Separation of Concerns through Categories

The purpose of categories is to divide the structure of business objects into clearly defined concerns. Organizational charts sometimes mix different structural aspects into the same diagram, which can lead to problems in an ERP implementation (Jean-Paul Smets, Nexedi SA, personal communication, April 18, 2010). Therefore, in ERP5 each aspect of the organizational structure can be defined in a distinct category. The combination of several categories can then define a new aspect. A position in a company, is for example defined per default by group, site and function. But the ERP adaptor can choose any combination of categories to define positions in the company. Also, the position is clearly separated from grade and salary level, which are defined in its own respective categories.

This separation of concerns also increases flexibility in implementing an organization's structure. The site category, for example can define the location for departments or business units of an organization. For that purpose, site is used together with the group and function category. To denote for example that the marketing division of company “c” is located at the office in Dresden, an organization is created for the department, which is assigned to the group “c”, to the site “dresden” and to the function “marketing”.

3.5 Configuration Case

To better understand the process of category configuration, an exemplary configuration was conducted for Aurora System, a small Independent Software Vendor (ISV), that wants to adopt ERP5. The configuration was lead by a series of questions which are derived from an exams template for students to provide example configurations based on the questions (see Smets & Honoré, 2010).

This chapter presents those questions whose answers directly lead to the configuration of particular categories. The fully replied questionnaire and the tables resulting from the configuration are stored on the compact disc attached to this thesis (see appendix A: Content of the Compact Disk, p. VII). The tables can be imported into ERP5. The following section explains, how the answers to the questionnaire lead to the configured categories, though only the most important categories are described. To view the whole set of configured categories, see the printed version of the tables in appendix E (Categories of the Configuration Case, p. XIV).

Question: What does the implementation field sell, offer or produce?

Answer: Aurora Systems develops and sells Kallimachos, a library management software. The program is highly specialized for use in school libraries, therefore the target marked is quite narrow. The software is sold as school wide licenses and update licenses. The price depends on the extend of needed functionality. Accompanying Kallimachos, hardware (barcode scanners), expandable items (barcode labels, protective film) and services (support, custom function programing) are sold.

Aurora Systems offers a second product line, called meine-schulbibliothek.de Under this brand Kallimachos hosting services are rented on a yearly basis, also accompanied by above mentioned products and services.

Consequences for category configuration: The question was centered on *product line* category. From the answer a hierarchy can be deduced with two or three levels, depending on the wishes for detail. The top level of the hierarchy consists of four main product lines:

- kallimachos,
- meine-schulbibliothek.de,
- expandabel items and
- hardware / scanner

The Kallimachos line includes several types of services:

- kallimachos / license (regular licenses),
- kallimachos / update (update licenses),
- kallimachos / support and
- kallimachos / programming (service for programming custom functions)

Licenses can be treated as services in ERP5 because they don't have any stock. meine-schulbibliothek.de also contains support and programming services. Additionally it adds a category for a "hosting" service to provide the Kallimachos program to schools on the Internet.:

- msbde / hosting
- msbde / support and
- msbde / programming

The expandable items category contains barcode and protective film as sub-categories. For barcode scanners, a differentiation is made between CCD scanners and laser scanners:

- expandable / barcode
- expandable / film
- hardware / scanner / ccd
- hardware/ scanner / laser

The answer also determines the addition of "user" as a *role* category. It will be assigned to all persons which use the meine-schulbiblothek.de SaaS. Since persons can be assigned to multiple roles, some users might be clients that actually pay for a hosting service. Other users might be sales prospects, that use the service for testing purposes.

Question: What does the implementation field purchase, recycle, receive or use?

Answer: Aurora systems purchases hardware in form of barcode scanners which are configured and sold for use with the Kallimachos software. Further, expendable items like labels, transparent film, thermo transfer ink ribbon to print barcode labels, CDs and office material are purchased regularly. The company uses computers, office printers and a special barcode printer.

Consequences for category configuration: The answer to this question resulted in only one additional category for product line:

- expandable / ink_ribbon

Other expandable items and office material weren't defined as special categories. They will be assigned to the expandable items category respectively to the hardware category.

Question: Who are the contacts of the implementation field?

Answer: Aurora System's clients are all kind of non-academic schools. The contacts are professors, secretaries and school directors. Sometimes regional public administrations or towns can be clients too, if they buy licenses for many of their schools together. On the supplier side, contacts are sales agents.

The most important contacts are public Länder administration organizations, like the Bavarian state institute for education research or the Bavarian state library who advises schools and highly influences their decision on which library management software to buy. These organizations usually have many satellite offices distributed above Bavarian cities. The agents in these offices are important "multipliers" because their opinion about library software largely influences the schools in their region.

Consequences for category configuration: This answer gives clues about the following categories:

- role,
- region,
- group,
- function and
- activity

An extra category is added to *role*: "multiplier" is a category on its own because the concept cannot be described with the standard role categories like client or media.

In the *region* configuration, the category "germany" is further divided into a subcategory for each Bundesland. Germany is Aurora System's main marked. The most important distinction is between different Bundesländer, because there are different school laws and schools administrations in each Bundesland. Most Aurora System's schools are located in Bavaria, so it is further distinguished between different Regierungsbezirke as it is important to Aurora System's clients to cluster client schools per region to give references and cluster schools for organizing training sessions.

Normal school clients don't need an entry in *group* configuration. It is only necessary to model larger organization with subsidiaries. Therefore, several categories for educational institutes that are important multipliers for Aurora Systems have been added as part of the Bavarian ministry of education and science. Also groups for cities, like Munich have been added, where the city administration participates as a client.

For *function*, several categories have been added to denote the function of client contacts, for example

- education / professor
- education / agent (school secretary)
- education / manager (school director)

Question: What are the typical skills and initial training of the staff?

Answer: The most important skills are IT skills, management skills and "sales and distribution skills". IT skills are needed in the areas of programming (application- web, and system programming, languages), system and web administration. Management skills are most important in the areas of marketing and especially communication which requires deep knowledge about our software and what the different stakeholders demand. Sales and distribution skills are rather specific to our sales and distribution process. It's required to know enough about our products to advise a client. Furthermore it demands the ability to configure our software and a barcode scanner as well as to use the label printer. The "user support" skill is product specific and requires only little IT knowledge (system administration support is apart from general user support) Soft skills are also required for marketing and communication, English language skills for software development.

The *activity* hierarchy resulting from this answer is quite detailed with different education, IT and governmental activities. In large parts, they form a subset of the standard activities.

Consequences for category configuration: This answer resulted in a quite detailed *skill* hierarchy, especially for IT skills with different programming techniques and operating systems. Most skills are also applicable to other ISVs. A highly specific skill is "sales / distribution / barcode" which defines the skill to configure a barcode scanner for use with the Kallimachos program. This skill can also be assigned to some client contacts, for example system administrators of schools or cities, so clients can be advised, who in their region they can ask for help with their barcode scanner.

Question: Please provide an example of management area or of business process which the implementation field is handling in a way which it considers itself as being good or successful. Explain what reasons make this business process or business area successful.

Answer: A successful business process of Aurora Systems is the sale order and distribution process, which is quite streamlined and quick and at the same time highly flexible. It's based on the distinction between standardized and flexible workflow components. Standardized parts in the sales process are conducted by the sale agent at the Höchberg site. It consists in barcode scanner configuration, label printing, and product packaging. Some sale orders are purely standardized like a barcode label sale order and are only treated by the Höchberg site. The client-advising and software configuration part is conducted by the Dresden site and is very flexible. There is a highly detailed order form and clients are supported in their decision of what functionality to buy, so the software can be afforded by very small primary school libraries (200 books) as well as serve the need of big high school libraries (20.000 books) Custom function programming and foreign data integration can be included in the software configuration. The configured software is then packaged and sent by the Höchberg site.

Consequences for category configuration: The answer for this question gives a hint to configuration of the *site* category. Two sites are configured: Höchberg and Dresden.

Question: Please provide an example of management area or of business process which the implementation field is handling in a way which it considers itself as being poor or wrong, and which could be improved according to him or her. Explain from what point of view the management area or business process is currently not well implemented.

Answer: The CRM and pre-sales process is still very poor. There is no management of leads and prospects. Prospects are either not contacted again, if the initial contact doesn't lead to a sell or they're only contacted on an irregular basis. Customers are not informed about product updates regularly and not asked, if the product is used successfully. The market is well defined, and it is possible to get contact information about all our leads, but marketing instruments are not used methodically. Communication with "multipliers" which influence the opinion of our leads is not managed and far to irregular.

Consequences for category configuration: This answer leads to the addition of the categories "sales lead" and "sales prospect" to *role* and affirmed the importance of the "multiplier" role. It also reinforced the need for a detailed definition of the *region* categories to cluster all potential clients per region and to regular contact multipliers in every region in Germany.

For *publication section*, this answer means, that marketing is important for Aurora Systems, so a specific category for marketing documents was added, with subcategories for the two main product line kallimachos and meine-schulbibliothek.de to categorize marketing material accordingly

The only categories that were not yet mentioned are nationality and grade. For *nationality*, simply the standard categories with all nationalities were used, although it could be reduced to European German speaking nationalities. Three *grade* categories were defined: employee, trainee and associate.

4 Automating Category Configuration

The previous chapter concluded that the *configuration* type of package tailoring is most likely suitable for automation. It presented *category configuration* as a specific ERP5 configuration option. This chapter describes how category configuration can be automated. First, the procedure to automate the configuration of a selected category is presented. The procedure is generalized to automate any configuration option which is suitable for automation. This is followed by a discussion of different automation approaches. A long-term strategy for managing questions and collecting configuration data is explained. Finally, considerations about the automation of the configuration of selected categories are given.

4.1 Automation Procedure and Information Sources

The automation of category configuration is conducted by following the procedure outlined in Figure 4 (Automation procedure and information sources, p. 39). The aim is to develop a general automation procedure which can be applied on different configuration tasks. Therefore the specific procedure for the automation of the category configuration is abstracted. The purpose of this abstraction is to support future investigations like the automated configuration of further categories as well as any configuration option which is suitable for automation.

To prepare the automation the following apsects of configuration options are analyzed:

1. the effect of the configuration option to the functionalities of the ERP system,
2. the structure of the values accepted by the configuration option (for example a particular data type),
3. possible and realized configurations and their reasons, and
4. the applicability of different automation approaches to automate the configuration option.

The first two points aim at understanding the configuration option itself from a technical point of view. It requires knowledge on the ERP system to understand the effect of the configuration options to its functionalities. The structure of the values accepted by the configuration options defines the theoretical possible values. For simple configuration options the values can be defined by the data type of the option, for example string or integer. Category configuration in ERP5 is an example for a more complex structure, in this case a hierarchy of objects which have certain attributes like id, title etc. (see chapter 3.4: Category Configuration, p. 26).

The purpose of the third aspect is to understand which value would be set for the selected configuration option to fulfill a particular business requirement. Thus, possible values for the configuration option are considered from a business point of view instead of a technical point of view. The business requirement that leads to set the configuration option to a particular value can be

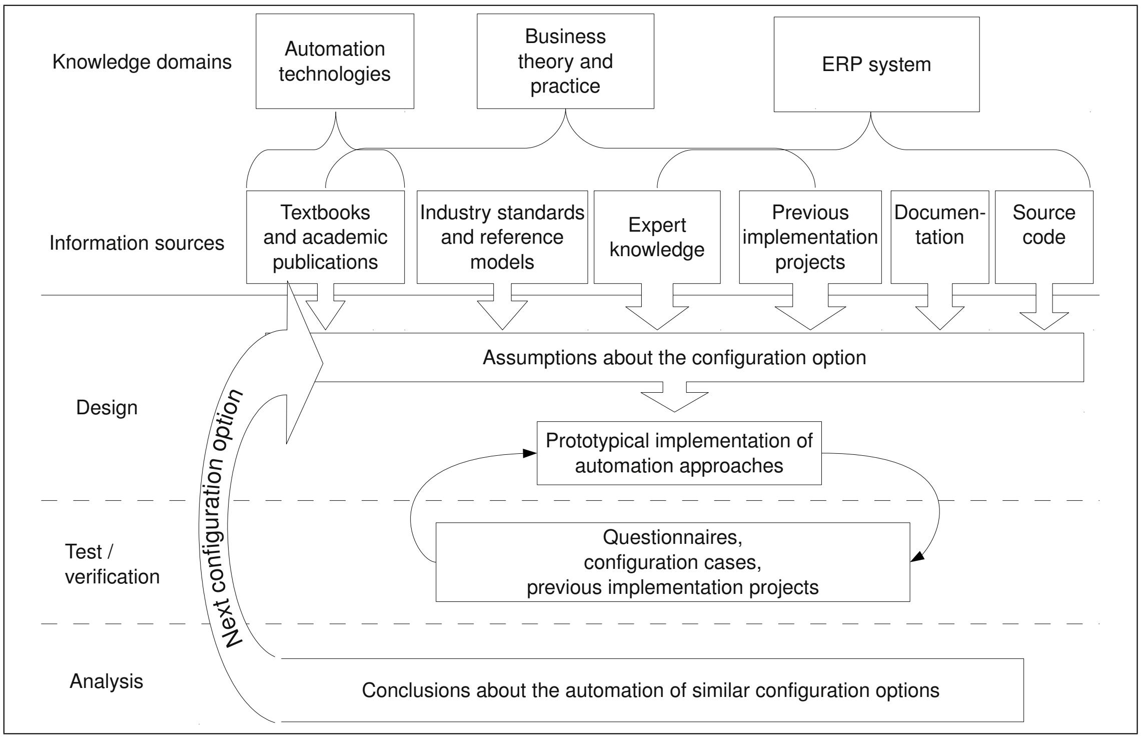

Knowledge domains
Automation technologies
Business theory and practice
ERP system
Information sources
Textbooks and academic publications
Industry standards and reference models
Expert knowledge
Previous implementation projects
Documen-tation
Source code
Design
Assumptions about the configuration option
Prototypical implementation of automation approaches
Next configuration option
Test / verification
Questionnaires, configuration cases, previous implementation projects
Analysis
Conclusions about the automation of similar configuration options

Figure 4: Automation procedure and information sources

understood as the "reason" why the value is set. Each configuration value and the corresponding reason form together a configuration case for the particular configuration option. Depending on the automation approach it might be neccessary to narrow the set of supported configuration cases.

The fourth item consists in describing automation approaches and to analyze their applicability to automate the selected configuration option. The purpose is to select suitable automation approaches for the prototypical implementation in the design phase of the automation process.

To conduct the described analysis, information is gathered from different information sources belonging to different knowledge domains (see Figure 4). Information about *the effect of the configuration to the ERP system* and about *the structure of the accepted configuration values* originate from knowledge on the ERP system itself. The relevant information sources are:

- expert knowledge of developers of the ERP system and consultants who implement it,
- configuration values of previous implementation projects and their documentation,
- documentation of the ERP system including:
  - architectural documentation and design documents,
  - technical documentation,
  - implementation documentation and
  - user documentation,

- source code and screen masks of the ERP package.

The results of this technical analysis regarding category configuration are presented in chapters 2.4.3 (Meta Planning with Categories, p. 11), 3.2 (Tailoring Options in ERP5, p. 18) and 3.4 (Category Configuration, p. 26). However, the latter is not purely technical, but also contains some information from an ERP5 adaptor point of view.

The analysis of *possible and realized configurations and their reasons* requires theoretical and practical business knowledge as well as knowledge on realized configurations of the selected configuration option. Theoretical business knowledge is needed to anticipate possible configuration cases. Past configurations are analyzed from the point of view of the organization which adopted the ERP system to find out which configurations were conducted for which reasons. This requires practical knowledge on implementing the particular ERP system. Therefore the relevant information sources are located in the domain of business theory and practice as well as the domain of the particular ERP system. They are:

- text books and academic publications,
- industry standards and reference models,
- configuration values of previous implementation projects and their documentation, and
- expert knowledge of developers of the ERP system and consultants who implement it.

Results of this analysis of category configuration from an implementation point of view are included in the chapters 3.4 (Category Configuration, p. 26) and 5.1 (Decision Tree Based Automation of Site Configuration, p. 49).

Reference models are usually associated to the IS domain. It is difficult to use general reference models of IS literature for automating ERP5 configurations (see chapter 5.1: Decision Tree Based Automation of Site Configuration, p. 49). The reason is that ERP functionalities are implemented in a more general way in ERP5 than in other ERP systems. Still, reference models can contain valuable information about different business requirements, therefore they are associated in Figure 4 to the business knowledge domain.

Expert knowledge on category implementation has been gathered by:

- an initial expert interview,
- personal communications with ERP developers and consultants and
- expert questionnaires to verify the initial assumptions and the prototypical implementations.

The interview has been conducted with Jean-Paul Smets, Nexedi SA, on November 2, 2009 before starting the investigations. It led to a rough understanding of ERP5 category configuration and resulted in first ideas for possible automation approaches based on artificial intelligence. The draft version of the prototype has been verified using an expert questionnaire. Based the answers to this questionnaire, the final version of the tree was designed (chapter 5.1: Decision Tree Based Automation of Site Configuration, p. 49). The assumption about the categories *group*, *region* and *role* were also verified using expert questionnaires. All questionnaires were replied by the initiator of ERP5, Jean-Paul Smets, and by an experienced ERP5 consultant, Tierry Brettnacher, both from Nexedi SA. The interview and all questionnaire replies are stored on the attached compact disc (the interview in audio form; see appendix A: Content of the Compact Disk, p. VII).

Expert knowledge can also be generated by the researcher himself during the automation procedure. Therefore the category configuration for an exemplary configuration case is conducted during this thesis (see chapter 3.5, p. 33). In addition, five previous configuration projects were analyzed, particularly for their implementation of site configuration (chapter 3.4: Category Configuration, p. 26).

The analysis of the *applicability of different automation approaches* requires knowledge of technologies that can be used for automation. The in this thesis discussed and implemented approaches for automating category configuration are based on concepts associated to the domain of artificial intelligence, in particular knowledge engineering and machine learning. Therefore, information sources are mainly textbooks and publications in this field.

The *design* and *test / verification* phases of the automation procedure are roughly based on the design as a search process (Hevner et al., 2004, pp. 88–90). The information gathering and analysis described above leads to assumptions about the selected configuration option. Based on these assumptions an automation approach is selected and implemented in form of a first prototype.

The prototype is then verified, preferably with the help of persons with expert knowledge on the mentioned knowledge fields, like ERP system developers and consultants. Expert questionnaires can be used to validate the assumptions on which the prototype is based. Exemplary configuration cases can help to test the prototype. The prototype can also be tested with past implementation projects by comparing the results of the automatic configuration with the original configurations.

The outcome of the verification phase can be either:

- the rejection of the automation approach which leads to a new iteration in the design phase to select a different approach and implement a new prototype,
- the validation of the approach with the correction of false assumptions or proposals for improvements which leads to a new iteration in the design phase to implement the improvements based on the existing prototype or
- the validation of the approach and the validation of the assumptions, which leads to the analysis phase.

The purpose of the *analysis* phase is to draw conclusions from the verified automation approach and its verified implementation for the automation of other, similar configuration options. These conclusions lead to new or changed assumptions about the other configuration options to automate.

In the case of category configuration this means that the successful test of the final decision tree of the site category leads to the conclusion that categories which have a similar bounded solution space can be also automated with the decision tree approach.

The automation process is fully applied to automate the configuration of the *site* category (chapter 5.1: Decision Tree Based Automation of Site Configuration, p. 49). Prototypical implementations are also conducted for the *role* category and the *product line* category. They are automated using the machine learning approach. An initial test with the training sets shows the applicability of the approach to category configuration. The test / verification phase can only be applied partly to these categories, because more data to learn the classifiers has to be collected first (chapter 5.2: Supervised Learning Implementation, p. 51). To prepare the implementation of the prototypes in ERP5, the ERP5 Artificial Intelligence Toolkit (EAT) is implemented as a set of three ERP5 modules. With EAT, the designed site decision tree is directly implemented in ERP5. EAT also supports the collection of sample data for the machine learning approaches, so that the test / verification phase can be fully applied to the *role* and *product line* prototypes in future research.

The presented procedure covers automation of suitable configuration options from a prototyping point of view. It covers the phases conducted in this thesis. The next step is to convert the designed prototypes into a production system. Then, real ERP adopters can reply to the configuration questions and work with the automatically configured instances. This will allow evaluation of the investigated approaches as demanded by the design-science research guidelines (Hevner et al., 2004, pp. 85–87).

4.2 Automation Approaches

Two approaches to automate category configuration based on artificial intelligence are introduced in this section. Both approaches share the idea to ask the adopting organisation a list of questions and then automatically generate the category configuration based on the given answers. The first approach is based on knowledge engineering. The rules that decide which categories are configured are defined in a *decision tree*. The second approach is based on machine learning. A *classifier* learns from training data how to classify a set of answers to a set of categories.

4.2.1 Knowledge Engineering with Decision Trees

The idea of the knowledge engineering approach to automate category configuration is to manually define a set of rules. These rules encode expert knowledge on how to configure categories based on answers to a list of questions. A decision tree is a form to represent these rules.

Decision trees can be used for example for a classification problem. In this case, the input to the decision tree is a set of attributes and the tree returns a decision. Each node of the tree corresponds to a test of the value of one of the input attributes and each branch from the node is labeled with one of the possible return values of the test. Each leave node in the tree specifies the decision that is returned if the leave node is reached. In the case of Binary classification, each leave node would specify either “true” or “false”. (Russell & Norvig, 2003, p. 653)

In the case of category configuration, a decision tree decides which categories are to add to the ERP instance. The attributes, that are the input to the decision tree correspond to the questions asked to the ERP adopter. The value of an attribute corresponds to the answer to a question. A decision specified in a specific leave node corresponds to a set of categories.

The categories cannot always be specified in advance. For example in site configuration, the knowledge engineer cannot know how an ERP adaptor wants to name the sites of his organization. Therefore the values specified in the leaves of the decision tree have to be generated dynamically.

Also, the attributes are not known from the beginning. The automatic configuration system first has to ask questions to the ERP adaptor to gather the answers which would be the input for the decision tree. This introduces the risk of asking questions that might not have been necessary because the nodes which would test the related answers might never be reached.

Therefore the concept of decision tree is adapted to meet the specific requirements of automating category configuration in the following ways:

- instead of asking the questions beforehand, they are defined by the nodes of the decision tree;
- instead of specifying decisions statically in the leave nodes, logic which dynamically generates categories is defined in the branches.

The decision trees guide the user through a list of questions. Each node in the tree represents a question. Questions can be generated dynamically to support slight alterations of the same question. Rules define, which branch in the tree is followed after answering the question. These rules usually decide based on the value of the answer, though they can access any previous answer and any information in the ERP system, too. The branches contain logic that performs the actual category configuration and therefore represents the decision of the tree. This logic is evaluated at branch activation. The branch then points to the next node which contains the next question. If the node is a leave, configuration is finished. Chapter 5.1 (Decision Tree Based Automation of Site Configuration, p. 49) presents a decision tree which implements this approach. Chapter 5.3 (Implementation of the ERP5 Artificial Intelligence Toolkit, p. 55) presents the implementation of the infrastructure with which this kind of decision trees can be implemented in ERP5.

This concept of decision trees is not as narrowly defined as for example the classification decision trees mentioned above. The actual features of a particular automation decision tree depend on the expressions, which decide the branch to activate and on the configuration logic defined in the branches. The advantage of this approach is that it is very flexible and allows to construct rather small trees (the number of nodes is usually equal to the number of different questions).

This makes this approach favorable for knowledge engineering as the decision process follows the questioning process. It allows an ERP consultant to build the decision tree in a similar way in which they interview their clients to configure an ERP. Although the tool is easy to handle, knowledge engineering experience is necessary to define decision trees.

The drawback of the knowledge engineering approach is the "*knowledge acquisition bottleneck*" (Sebastiani, 2002, p. 9). The decision tree must be manually defined by an ERP implementation expert with the aid of a knowledge engineer. Extensive knowledge about the ERP system, the specific configuration option and about related requirements of businesses is neccessary. Possible configuration cases have to be anticipated. For many categories it's impossible to cover all cases. The configuration cases supported by the decision tree have to be narrowed.

Therefore the next chapter introduces a second approach which allows an automation system which can learn new configuration cases and evolve continually.

4.2.2 Classification with Machine Learning

To apply the machine learning approach, the problem of automatic category configuration is viewed as a *classification* problem. A category configuration questionnaire is replied by an ERP adopter. The set of answers which is produced in this questioning process has to be classified to a set of categories which together form the category configuration.

The problem is similar to the problem of text classification (also called text categorization). In text categorization a document has to be classified to categories, such as classifying email to the categories "Spam" and "Not Spam". In the case of automatic category configuration, the documents to be classified are just sets of answers to a questionnaire. Therefor the following formal definition is very similar to the definition of text categorization by Sebastiani (2002, p. 3).

The problem can be defined as assigning a Boolean value to each pair $\langle a_j, c_i \rangle \in A \times C$, where A is a domain of answer sets and $C = \{c_1, ..., c_{|C|}\}$ is a set of predefined categories. A value of T assigned to $\langle a_j, c_i \rangle$ indicates a decision that the configuration based on the answer set a_j should include the category c_i. A value of T indicates a decision that the configuration should not include the category c_i. More formally, the task is to approximate the unknown target function $\breve{\Phi} : A \times C \to \{T, F\}$ (that describes how answer sets ought to be classified to be classified to categories) by means of a function $\Phi : A \times C \to \{T, F\}$ called the *classifier* such that $\breve{\Phi}$ and Φ coincide as much as possible.

Three kinds of classification can be distinguished (Sebastiani, 2002, p. 3):

- multi-label classification: answer sets are assigned to more than one category
- single-label classification: each answer set is assigned to exactly one category
- binary classification: each answer set $a_j \in A$ must be assigned to either c_i or to its complement $\bar{c_i}$.

The case of automating category configuration is a problem of multi-label classification, because an answer set (even if it contains only one answer) mostly leads to the configuration of more than one category. Sebastiani (2002, p. 3–4) explains, that an algorithm for binary classification can also be used for multi-label classification:

“one needs only transform the problem of multi-label classification under $\{c_1, ...|C|\}$ into $|C|$ independent problems of binary classification under $\{c_i, \bar{c_i}\}$, for $i = 1, ..., |C|$.”

Therefore, the prototypical implementations of the machine learning approach described in chapter 5.2 (Supervised Learning Implementation, p. 51) transforms the problem of assigning multiple categories to an answer set into multiple independent problems of assigning a particular category or not assigning this particular category. Given for example three product line categories “fish”, “computer” and “packaging”, the prototype solves the three binary problems of assigning:

- ‘fish” or “not fish”,
- “computer” or “not computer” and
- “packaging” or “not packaging”.

However, this requires that categories are stochastically independent of each other, i.e. that for any c', c'' the value of $\breve{\Phi}(d_j, c')$ does not depend on the value of $\breve{\Phi}(d_j, c'')$ and viceversa; this is usually assumed to be the case in text categorization (Sebastiani, 2002, p. 4). It is often not the case in ERP5 category configuration because of the hierarchical structure of categories. For the case of Web pages Sebastiani (2002, p. 8) suggests that the hierarchical structure of the category set can be used for example by de-composing the classification problem into a number of smaller classification problems, each corresponding to a branching decision at an internal node. This might be appropriate for ERP5 category configuration, too: if the automated system is classified as “not product_line / computer”, there could be a decision not to try to classify “product_line / computer / hardware”. Though this is not yet implemented in the prototypes; they are configured for flat categories only at the moment of this writing.

Using binary classification for automating category configuration, $C = \{c_1, ..., c_{|C|}\}$ is viewed as consisting of $|C|$ independent problems of classifying the answer set in A to a given category c_i, for $i = 1, ..., |C|$. A classifier for c_i is then a function $\Phi_i : A \times C \rightarrow \{T, F\}$ that approximates an unknown target function $\breve{\Phi} : A \times C \rightarrow \{T, F\}$.

The knowledge engineering approach to the classification problem is to build a classifier which consists of a manually defined set of rules, one per category, which classify the documents (the answer sets in this case), see Sebastiani (2002, p. 8). The decision tree approach presented in the previous chapter is similar, with the difference that it does not operate on an already existing document, but the document (the answer set) is constructed during the evaluation of the rules.

The idea of the machine learning approach is not to construct a classifier but an automatic builder of classifiers, the learner. Many learners are available off-the-shelf. The open source orange toolkit (http://www.ailab.si/orange) used in the binary classification prototype (chapter 5.2.2: Automation of Role Configuration with Binary Classification, p. 52) offers for example a Naïve Bayes classifier, k-nearest neighbours, a tree learner, support vector machines and rule learners amongst others.

Both prototypes presented in chapter 5.2 (Supervised Learning Implementation, p. 51) use the Naïve Bayes classifier. It is based on the inductive construction of ranking classifiers, which for a category $c_i \in C$ consists in defining a function $CSV_i : A \rightarrow [0,1]$ which, given an answer set a_j returns a *categorization status value* for it. It is a number between 0 and 1 which indicates that $a_j \in c_i$. Category-ranking then works by ranking the CSV_i scores of a given answer set a_j for the different categories in $C = \{c_1, ...c_{|C|}\}$ (Sebastiani, 2002, p. 21).

The Naïve Bayes approach is a probabilistic classifier. Given is a set T of binary or weighted terms that represents the answer set. In the case of the prototype presented in chapter 5.2.1 (Automation of Product Line Configuration with Text Classification, p. 51) these are the weighted terms of a single answer. In the prototype presented in chapter 5.2.2 (Automation of Role Configuration with Binary Classification, p. 52) the discrete answers of the questions are converted into binary weighted terms, where the term corresponds to the identifier of the question. The weights are "y" which corresponds to weight of 1 (present) and "n" which corresponds to weight of 0 (not present). A probabilist classifier views $CSV_i(a_j)$ in terms of a probability $P(c_i|\vec{a_j})$ which is the probability that an answer set represented by the vector $\vec{a_j} = \langle w_{1j}, ..., w_{|T|j}\rangle$ of binary or weighted terms belongs to c_i (Sebastiani, 2002, p. 22). The probability is computed by an application of Bayes' theorem given by

$$P(c_i|\vec{a_j}) = \frac{P(c_i)P(\vec{a_j}|c_i)}{P(\vec{a_j})}$$

The estimation of $P(a_j|c_i)$ is problematic, since the number of possible vectors is too high. To alleviate this problem the Naïve Bayes classifiers make the assumption that any two coordinates of the answer set vector are statistically independent of each other. This *independence assumption* is encoded by the equation

$$P(\vec{a_j}|c_i) = \prod_{k=1}^{|T|} P(w_{kj}|c_i)$$

This assumption has to be taken into account in constructing the questionnaires. The category configuration should be split into multiple questionnaires. Each answer set which corresponds to a questionnaire should contain only statistically independent answers. The categories are then classified to each answer set independently. The category configuration is then the union of all independently classified categories.

The prototypes can be changed to use other classifiers easily. Future research should include studies to identify the most favorable classifier for the case of category configuration.

Since the learner is available off-the-shelf all that is needed is to *train* the learner. Therefore a training set has to be classified manually. The learner constructs from the training set a classifier automatically. In the case of category configuration, the training set consists of multiple answer sets that are each assigned to a set of categories. This can be done for example in letting students create example configurations and answer a questionnaire. The answer sets and the related example configurations (corrected by the professor) together make up the training set. To facilitate this for the case of ERP5, chapter 5.3 (Implementation of the ERP5 Artificial Intelligence Toolkit, p. 55) presents three ERP5 modules which are used to construct questionnaires and collect answers sets from students together with example category configurations. Once enough training data is collected, prototypes can be evaluated and evolved into an automatic configuration system using the machine learning approach. The same questions like the ones defined in the questionnaires can be used to implement the questionnaires and decision trees which will be asked to an ERP adaptor for the automatic configuration.

The advantage of the machine approach over the knowledge engineering approach is that it is easier to manually classify multiple answer sets than to build and tune a set of rules or a decision tree "since it is easier to characterize a concept extensionally (i.e. to select instances of it) than intensionally (i.e. to describe the concept in words, or to describe a procedure for recognizing its instances)" (Sebastiani, 2002, p. 10).

The SaaS model of TioLive also favors the machine learning approach. Once an initial training set is available, the first instances for new TioLive adopters can be configured automatically. These initial configurations can then be improved manually by human consultants. The improved instances together with the answer sets provided by the TioLive adopters are then the new high-quality training sets. Therefore, the automatically constructed classifier continually evolve to support different and new business requirements to category configuration.

Still, a problem also remains with the machine learning approach: Which are the right questions to ask? Therefore a "pure" machine learning approach is not possible. Asking the right questions to ERP adaptors is part of the knowledge of ERP consultants. To tackle this problem, chapter 5.3 (Implementation of the ERP5 Artificial Intelligence Toolkit, p. 55) presents a ERP5 module which serves to collect different kinds of questions. The module contains a validation workflow so that questions that have been proved suitable can be validated and obsolete questions can be invalidated. Once training data is available, the different questions can be tested for their suitability for category configuration. During the investigations, a number of questions have been collected. They are presented in the different prototypes, see chapter 5 (Prototypical Implementation, p. 49). These questions have already been added to the question collection module.

A disadvantage of the current prototype of the machine learning approach is that all questions will be always asked, also if some questions would not make sense based on previous answers. This is related to the *independence assumption* and the hierarchic structure of categories. The problem can be solved by splitting the questionnaires into independent questionnaires. A combination of the machine learning approach with the decision tree approach could also be a solution. After the automatic classification of a part of the answer set, a manually defined rule would define if the next set of questions is asked or not. Thoug, also this rule could be substituted by a learning classification.

A decision which approach to use should be done on a per category level. The successful implementation of the site decision tree, see chapter 5.2.1 (Automation of Product Line Configuration with Text Classification, p. 51) indicated that a decision tree is a suitable approach for categories having a similar low complexity, like *group*, *role*, *region*, *nationality*. Categories with a high number of possible values, like *product line*, *activity* and *publication section* should be more suitable for the machine learning approach. *Function* is still undecided, though test data should be collected for configuration of all categories, then the results of learning classification can be compared with the results of decision trees.

5 Prototypical Implementation

This chapter presents the prototypical implementation of the suggested approaches to automate category configuration. Three kinds of prototypes are presented: a decision tree, two machine learning code examples, and EAT, a toolkit consisting of three modules that supports to implement the discussed artificial intelligence approaches in ERP5.

5.1 Decision Tree Based Automation of Site Configuration

The implementation of the site decision tree was conducted following the automation procedure outlined in chapter 4.1 (Automation Procedure and Information Sources, p. 38). The final tree is an implementation of the approach described in chapter 4.2.1 (Knowledge Engineering with Decision Trees, p. 42). First, a draft version of a decision tree has been designed using the diagram creation program “dia” (`http://projects.gnome.org/dia`). It is based on the initial assumptions about the site category. These assumptions were taken from the initial information gathering as described in the automation process.

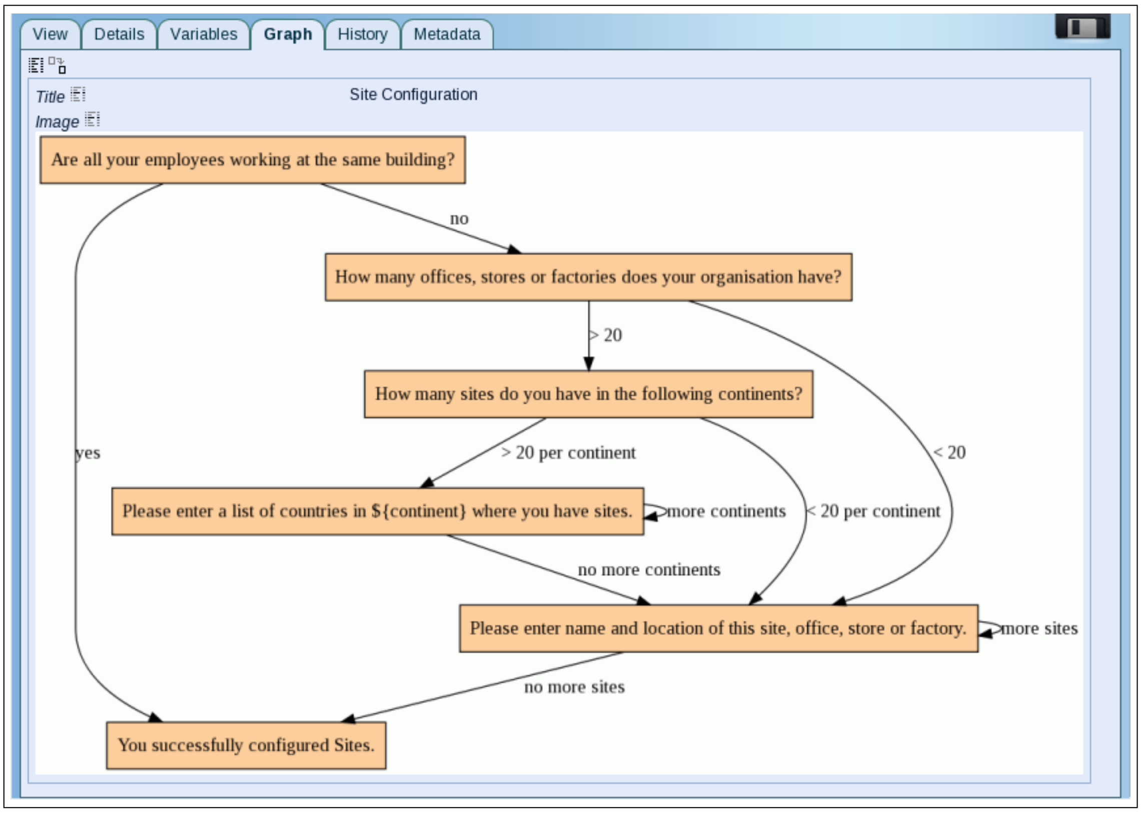
The draft version is shown in appendix D (Draft Version of the Site Decision Tree, p. XII). It consists of two parts. The first part configures a hierarchy of sites until the level of a particular building. It lets the user decide the aspect on which the hierarchy levels are built (for example ‘continent’, ‘country’ or ‘city’). Any number of hierarchy levels can be configured. The second part configures storage on the level below a particular warehouse. The implementation of the storage hierarchy was influenced by a reference model (Scheer, 1997, p. 138) and by storage technology literature (Hompel, 2007, p. 110).

To verify the decision tree and the initial assumptions, a questionnaire was created. It was replied by Thierry Brettnacher, Nexedi SA on April 12, 2010 and by Jean-Paul Smets, Nexedi SA on May 1, 2010. The replies are available on the attached compact disc (see A). The answers show that the decision tree approach is promising, but some of the assumptions about the site category and related ERP5 functionalities were not correct:

- site configuration should use different concepts than region category to avoid confusion,
- fine grained site hierarchy is too advanced for the case of SMEs,
- fine grained storage hierarchy is not needed for the current implementation of material flow in TioLive.

The verification reveals that the used reference model can not be applied directly to ERP5, because ERP5 models resource flows in a more general way than other ERP systems. The draft also tries to cover too many possible configuration cases. This leads to questions which are too abstract.

Therefore, a new decision tree was designed based on the answers to the questionnaire. The EAT


View
Details
Variables
Graph
History
Metadata
Image
Are all your employees working at the same building?
no
How many offices, stores or factories does your organisation have?
> 20
How many sites do you have in the following continents?
yes
> 20 per continent
< 20
Please enter a list of countries in ${continent} where you have sites.
more continents
< 20 per continent
no more continents
Please enter name and location of this site, office, store or factory.
more sites
no more sites
You successfully configured Sites.

Title Site Configuration

Figure 5: Site decision tree drawn by the EAT design tool

prototype (see chapter 5.3, p. 55) has already been available at the time of the redesign. Thus, the new site decision tree is implemented with the EAT design tool. Therefore it is an implementation of the decision tree concept described in chapter 4.2.1 (Knowledge Engineering with Decision Trees, p. 42). Figure 5 shows the final version of the site decision tree as it is drawn by the decision tree design prototype.

The final version has accomplished the task of category configuration with fewer and easier questions. It does not allow to define any kind of hierarchy, as the draft version did. But this is a good thing, because it forces a sensible site configuration and does not allow the ERP adopter to make any mistakes. It illustrates, how a single decision tree can fulfill two contradicting goals for automatic configuration: support as many configuration cases as possible and ask few questions in simple cases for quick configuration. In the simplest case, only one question is asked, and a default site is created. If the user has less than 20 sites, a flat list of site names is created, for more than 20 site hierarchy levels of continents and/ or countries are created, depending on the geographical distribution of sites.

The decision tree was tested with the Aurora Systems configuration case described in chapter 3.5 (Configuration Case, p. 33) and with the previous implementation project described in the “Site” paragraph in chapter 3.4.2 (Description of Selected Categories, p. 27).

The test results are displayed in table 4 on page 51. The table shows for each implementation project the number of configured sites, the number of non-equal questions asked during configu-

Implementation Project	Sites	Non-equal questions	Configuration covered
SANEF	1	1	Yes
Aurora Systems	2	3	Yes
NXD	6	3	Yes
MEDICENTRE	18	3	Partly
AUCKM	28	4	Yes
BAOBAB	>300	–	No

Table 4: Results of the site decision tree test

ration (not counting question repetitions), and if the particular configuration cases is covered. The results show that the site decision tree correctly configured every implementation project except BAOBAB and MEDICENTRE. The BAOBAB project with more than 300 sites is clearly out of scope of automatic configuration for SMEs. To fully support the hierarchy of the MEDICENTRE project, a question could be added that asks, if the adaptor has several buildings at the same site.

The site decision tree shows that the knowledge engineering approach is suitable for some categories. However the case of the MEDICENTRE project also shows the disadvantage of the knowledge engineering approach: The decision tree only supports the kind of configuration cases for which it has been designed. To add support for a new kind of configuration case, the tree has to changed. Therefore the next chapter presents two working examples which use the machine learning approach for automating category configuration.

5.2 Supervised Learning Implementation

5.2.1 Automation of Product Line Configuration with Text Classification

In chapter 4.2.2 (Classification with Machine Learning, p. 44) a machine learning approach to automate category configuration has been introduced. In the following, an example script is presented, which implements a special case of the approach: a textual answer to a single question is classified to a set of categories.

This case corresponds to the text classification problem from which the approach presented in chapter 4.2.2 is derived. A common example of text classification is the spam filtering problem where emails are classified into the categories “Spam” and “Not Spam”. Likewise, the example classifies a textual description of the business activity of a company to a set of *product line* categories.

The example consists of a Python script which can be found in the file “Prototypical Implementation/Machine Learning/text_classification.py” on the attached compact disk (see appendix A: Content of the Compact Disk, p. VII). The script does not implement the learner itself. It uses the implementation of a Naïve Bayes learner from Reverend (`http://divmod.org/trac/`

wiki/DivmodReverend), a Python library which, in addition to Python, is required to run the example. The script can be run with the command `python text_classification.py`.

The script first defines a list of three examples, which will be used as training data. Each example contains an id, a list of assigned product lines and a description. A fourth example is defined, which does not contain any product lines. These will be assigned when testing the classifier.

The task of classifying multiple product lines to a description is a multi-label classification problem. As explained in chapter 4.2.2, the approach is to transform the problem of multi-label classification into multiple independent problems of binary classification.

Therefore, the script code instantiates a new classifier for each product line defined in the training set. It then walks the training list and looks for each example, which product line it contains and which it does not. All classifiers are then trained with the phrase containing the description of the example and an assigned value. The value is either "y", if the example contains the product line corresponding to the classifier or "n", if it does not contain the product line.

After training the classifiers they are tested with themselves and with the test example. Of course the data does not justify as a sample, the purpose is to show the concept. The test results are printed to the command line. The output of the script can be seen in appendix B (Output of the Text Classification Example, p. VIII). It shows for each example the id, the originally in the training assigned product lines and the probabilities for "y" and "n" for each classifier. It can be seen that the classifier correctly estimated the categories of the training set. The "service" product line has always a probability of 100%, because every example in the training set was assigned to "service". The last entry is the test set which is classified to a sensible configuration which fits good to the description.

An Automation system built on this approach, could propose the classified product lines. The user would then remove falsely classified categories and add additional categories. This would lead to a new training set.

5.2.2 Automation of Role Configuration with Binary Classification

The previous section treated the case of configuring categories based on a textual answer to one question. This section presents an example script that classifies a set of *role* categories to a set of answers originating from multiple questions. The problem definition and solution approach has been described in 4.2.2. The answers are not textual, but defined as one or multiple discrete values out of a list of possible answers.

The script is located at "Prototypical Implementation/Machine Learning/binary_classification.py" on the attached compact disk (see appendix A: Content of the Compact Disk, p. VII). Per default, it uses a Naïve Bayes learner, too. This time the learner is provided by Orange (http://www.ailab.si/orange), a data mining and predictive modelling software suite that can be accessed from Python. The example can also be used with other learners provided by Orange, like k-nearest neighbours or tree learner amongst others. The example can be run with the command `python binary_classification.py`.

The script first defines a list of questions, which are described by the following fields:

- id: the internal identifier of the question
- title: the question as it is displayed to the user,
- values: a list of possible answers,
- type: either “selection” (only a single answer is accepted) or “multiple_selection” (multiple answers are accepted).

Four questions are defined in the question list:

```
{
    'id'        : 'not-for-profit',
    'title'     : 'Are you a not-for-profit organization?',
    'values'    : ('y', 'n'),
    'type'      : 'selection',
}, # might influence: role, function, activity, publication_section
{
    'id'        : 'sell-what',
    'title'     : 'What do you sell?',
    'values'    : ('products', 'services',),
    'type'      : 'multiple_selection',
}, # might influence: product_line, activity, function
{
    'id'        : 'sell-how',
    'title'     : 'How do you sell your products ?',
    'values'    : ('retail', 'online', 'distributor', 'other',),
    'type'      : 'multiple_selection',
}, # might influence: role, publication_section
{
    'id'        : 'client-types',
    'title'     : 'What types of clients do you have?',
    'values'    : ('consumer', 'business', 'administration',
                   'not-for-profit',),
    'type'      : 'multiple_selection',
}, # might influence: role, publication_section, product_line, activity
```

The commentary at the end of each question is a rough assumption, which categories could be influenced by the respective question.

The training data consists of four examples, each defined by an id, a set of answers and a set of assigned categories, for example:

```
{   'id': 'Aurora Systems',
    'answers':
    {
        'not-for-profit' : 'n',
        'sell-what'      : ('products',),
        'sell-how'       : ('online','other'),
        'client-types'   : ('not-for-profit',),
    },
    'categories':
 [
        'role/internal',
        'role/client',
        'role/supplier',
        'role/admin',
        'role/user',
        'role/saleslead',
        'role/salesprospect',
    ]},
```

A test set, containing two test examples is defined similarly (with the difference that the test set does not contain the ‘categories’ field).

The classification approach as described in chapter 4.2.2 consists in converting the questions into terms, so that each answer can be represented by a set of Boolean values, where “y” denotes the presence of the term and “n” denotes the absence of the term in the answer set.

For binary questions, like “Are you a not-for-profit organization?” the term is the id of the question, for example ‘not-for-profit’. Questions with more than one possible answer first have to be converted into a list of terms based on the possible answers. Thus, the question “What do you sell?” is converted into two terms ‘sell-what_products’ and ‘sell-what_services’. This conversion is conducted in the `gen_discrete_questions()` function. The answers are then converted into Boolean values “y” and “n” and associated to the respective terms in the `gen_discrete_answers()` function.

The converted training data is available in .csv files in the folder “Prototypical Implementation/Machine Learning/tables/” on the attached compact disk (see appendix A: Content of the Compact Disk, p. VII). The files can be imported into a spreadsheet application to view the data.

Similar to the text classification example described in the previous chapter, a classifier is then constructed for each *role* category by initializing the learner with the training data. The classifier are then tested with the training set and the test set. The test output can be seen in appendix C (Output of the Binary Classification Example, p. X).

As expected for this approach, all test data is correctly assigned according to the respective original role configuration. The two last entries are the test examples, which are also correctly assigned according to the given answers (The member and the public amdministration role are correctly assigned to not-for-profit organization as well as the clients and the sales-related roles are correctly not assigned).

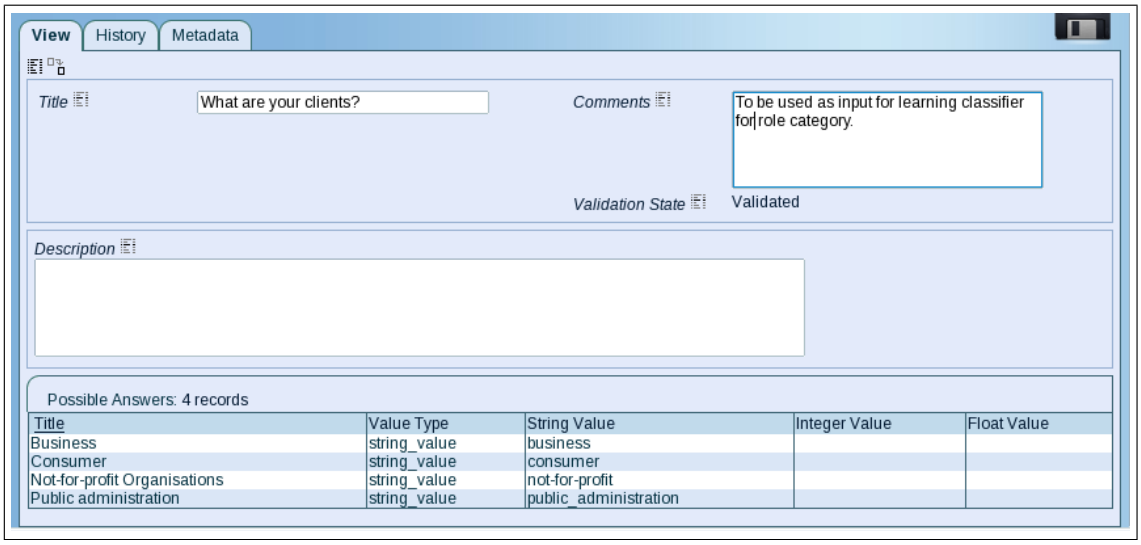
The results show the advantage of this approach: it is more predictable, because all possible answers are defined manually. This makes it also more feasible, if few training data is available. On the other hand, this approach has similar inconveniences as the decision tree approach, since the possible answers have to be configured manually which requires knowledge of ERP5 implementation. It is also less flexible to new unexpected configuration cases. Therefore, when collecting data, an additional field should be presented which allows to suggest a new answer value, if none of the predefined answers fits. With this addition, the approach might be a suitable compromise between the pure knowledge engineering approach and the text classification approach. Though it can also be applied in combination with the other two approaches.

5.3 Implementation of the ERP5 Artificial Intelligence Toolkit

Although the tests gave promising results in correctly assigning product lines and roles based on the answers they learned, the used data sets were very small. To validate the approaches, many example configurations and corresponding answers to the configuration questions are needed. Therefore, a prototype was implemented to directly define the questions in ERP5 and collect the answers together with a corresponding configuration. The idea is to give these questions to students that create a sample configuration for a small business. Together with the answers to the configuration questions, it will be the input to the learning classifiers. Once more learning data in form of answers and corresponding configuration data is available, the results can be compared to the results of using other learning algorithms like, -k-nearest-neighbor or Decision Tree Learners as classifiers.

The ERP5 AI Toolkit (EAT) is a collection of modules that help to implement artificial intelligence in ERP5. EAT consists of:

- a question management tool to create, collect and evaluate different types of configuration questions,
- a design tool that helps to create decision trees and questionnaires and
- an answer collection tool for data mining to collect answer data as input for learning classi-



Title: What are your clients?

Comments: To be used as input for learning classifier for role category.

Validation State: Validated

Description:

Possible Answers: 4 records

Title	Value Type	String Value	Integer Value	Float Value
Business	string_value	business		
Consumer	string_value	consumer		
Not-for-profit Organisations	string_value	not-for-profit		
Public administration	string_value	public_administration		

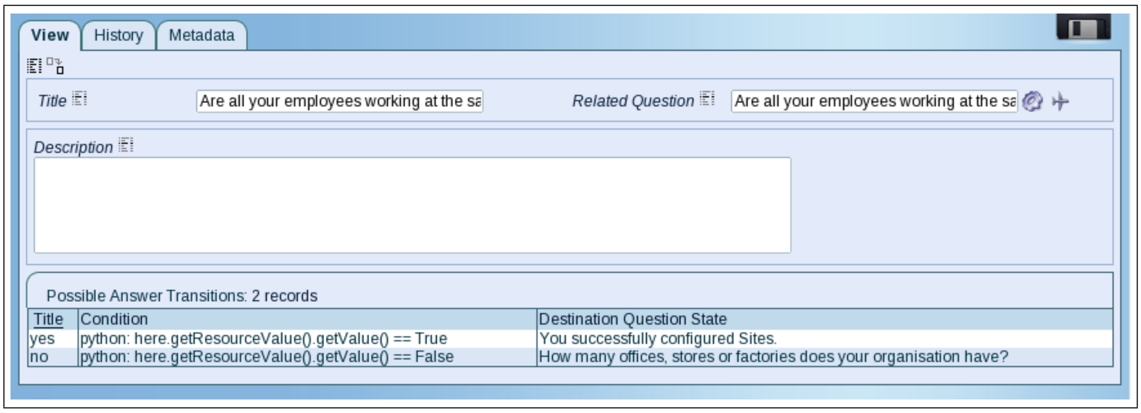
Figure 6: Question management tool showing a selection question

fiers and as a data basis for question evaluation.

Figure 6 shows the question management tool. It allows to create and collect different kinds of configuration questions, for example boolean questions, free text questions or questions with multiple possible answers. A special "document question" allows to upload documents in the questioning process, for example a spreadsheet with an ERP5 category configuration to connect the answers with a corresponding configuration as input for the learning classifiers. A validation workflow is attached to each question to allow question evaluation with the objective to identify redundant and overcomplicated questions for elimination or reformulation.

The design tool allows to relate questions to each other to create questionnaires and decision trees. Every node in the decision tree graph corresponds to a question. Different answer ranges can be defined for every question. Each leads to one of the next possible questions. These answer ranges represent the branches in the decision trees. They are defined by boolean expressions in the Python programming language and represent the condition under which the branch gets activated (see Figure 7). The expression is not bound to the answer of the last question, it can take any data in the system into account. The expressions of all arrows of a common node must be mutually exclusive, so that only one arrow can be activated at a time. Scripts can be defined for every arrow to take actions based on the user's input.

The graphs of the designed decision trees can be drawn automatically (see Figure 5 (Site decision tree drawn by the EAT design tool, p. 50). The prototype heavily reuses the ERP5 Workflow component, and could therefore be implemented in very short time. The decision trees which were created during this thesis are implemented with the EAT design tool to validate its functionality. Simple questionnaires, like exams, where all questions are displayed in a fixed linear order can be designed in the same way. In this case, every question has only one answer range, which is "All Answers", so the expression returns always "True". In a questioning process, the forms to



Title: Are all your employees working at the sa

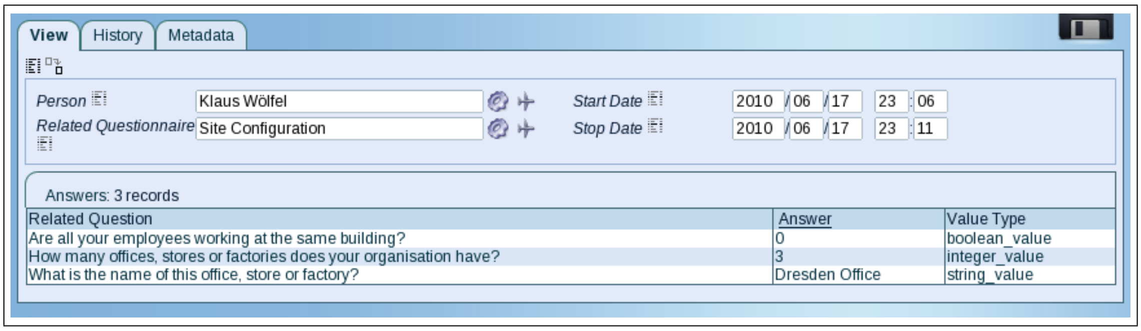
Related Question: Are all your employees working at the sa

Description

Possible Answer Transitions: 2 records

Title	Condition	Destination Question State
yes	python: here.getResourceValue().getValue() == True	You successfully configured Sites.
no	python: here.getResourceValue().getValue() == False	How many offices, stores or factories does your organisation have?

Figure 7: Design tool showing a question node related to a boolean question (The assigned answer ranges are shown with condition and destination node)



Person: Klaus Wölfel

Related Questionnaire: Site Configuration

Start Date: 2010 / 06 / 17 23 : 06

Stop Date: 2010 / 06 / 17 23 : 11

Answers: 3 records

Related Question	Answer	Value Type
Are all your employees working at the same building?	0	boolean_value
How many offices, stores or factories does your organisation have?	3	integer_value
What is the name of this office, store or factory?	Dresden Office	string_value

Figure 8: Answer collection tool showing an answer set for the site decision tree

display a particular question are implemented for each question type in the question management tool. The order, in which the questions are displayed, is defined by the corresponding decision tree or questionnaire. The answer collection tool (see figure 8) collects all answers of a questioning process. In configurations and exams, the effort of a question can be measured based on the time it takes to answer the question. This effort may then serve as an indicator for question evaluation.

The purpose of distributing the functionalities of the EAT is that questions can be used in a very general way in ERP5 serving different purposes. The same questions can be used for student exams as well as in a configuration decision tree. In the future, an initial configuration of TioLive should be accomplished automatically. If this basic configuration is then improved by a human consultant, the altered configuration together with the answers in the configuration process will be new input for the learning classifiers. Thus, the system will become smarter with every successful implementation.

6 Conclusion and Outlook

The research objective is to investigate the automation of ERP package tailoring for SMEs. This has been done on the basis of the open source ERP system ERP5. In chapter 2, the functions and technical architecture of ERP5 have been described, as well as the SaaS TioLive. The research questions are:

- Which tailoring options are most likely suitable for automation generally and in the case of ERP5 specifically?
- How can these ERP5 tailoring options be automated?

Configuration has been identified as the type of ERP package tailoring best suited for automation. The thesis shows on the basis of the chosen approaches and the prototypical implementation, how ERP5 category configuration can be automated.

From Brehm et al.'s categorization of tailoring types it is hypothesized that tailoring types with lower impact on the ERP system are more likely suitable for automation. From the application of ERP5's tailoring options to Brehm et al.'s typology and the analysis of the ERP5 implementation process it is concluded that many tailoring tasks in ERP5 can be accomplished by configuration and are therefore suitable for automation (see chapters 3.1 and 3.2).

The further investigations have been concentrated on the automation of category configuration. Therefore, this configuration option has been described from a technical point of view (chapter 2.4.3) and an implementation point of view (chapter 3.4). Eleven categories which are used in TioLive have been described. Previous ERP 5 implementations have been analyzed and an example configuration has been conducted to acquire practical knowledge of category definition (chapter 3.5).

After presenting a general procedure to automate suitable configuration options in chapter 4.1, two approaches to automate category configuration were discussed in chapter 4.2:

- knowledge engineering with a particular decision tree concept and
- a machine learning approach similar to text classification.

For both approaches, prototypical implementations have been implemented. A decision tree has been created, which automates the configuration of the *site* category (chapter 5.1. Its applicability has been verified by testing it with the configurations of previous ERP5 implementation projects and with the example configuration case. Two code examples have been created which implement the machine learning approach:

- a script which classifies *product lines* on the basis of a free-text answer and
- a script which classifies *role* categories on the basis of Boolean answers or the selection of

multiple possible answers.

Further, the ERP5 Artificial Intelligence Toolkit (EAT) has been created, a collection of three ERP5 modules which allows to collect sample data and implement decision trees directly in ERP5.

The prototypes and their initial verifications show promising results for automatic configuration of selected ERP5 categories. The prototypes has shown that decision trees are well suited for configuration options with narrowly defined value ranges. When configuration is more complex the knowledge engineering approach is more difficult. For these cases, classification based on machine learning seems to be a solution. The created code examples show that it is possible to configure categories based on text categorization with open questions as well as based on questions with discrete values. In particular the Machine Learning approach using multiple questions with predefined possible answers achieves good results. While these results are promising there are many aspects that need further investigation.

Most importantly, sample data has to be collected to train the learners. This would allow to refine the example code, to compare the results of the current implementations with the results of using other types of classifiers and to select the categories best suited for the machine learning approach. Different ways of using machine learning for category configuration should also be investigated. Existing documents of an organization, like invoices or packing lists, could be further sample data for text categorization. These documents hold value information for ERP5 configuration because of ERP5's document-centric approach to implement business processes.

The technological base for collecting sample data has been created with the EAT, but more questions have to be added to the question management tool. Once a large pool of configuration questions has been gathered, then the next step will be identifying interdependencies between questions in order to eliminate unnecessary ones. The overall objective is to find the smallest set of questions that covers the biggest range of possible configurations.

Also the decision tree has to be elaborated to cover more configuration cases and still stay simple for the user. The display of the questions in a questioning process should include automatically painted graphs for categories with multiple hierarchical levels. More sample data will allow to test the decision trees and refine them.

Converting the prototype into a production system with real ERP adopters replying to the configuration questions and working with the automatically configured instances will allow better validation of the investigated approaches. When automatic category configuration runs in production, other ERP5 tasks that include configuration, like

- security group definition,
- decision implementation,
- document implementation and

- integration

can be automated, too.

Then, an initial automatic configuration of ERP5 could be conducted for a small business based on a questionnaire replied by the owner or CEO. This basic configuration could then be improved by a human consultant. Using this approach with the TioLive SaaS, it could be offered as an integrated online service, named *Cloud Consulting*. Adopting an ERP this way would make it much more achievable for SMEs.

Reference List

Adam, F., & O'Doherty, P. (2000). Lessons from enterprise resource planning implementations in Ireland - towards smaller and shorter ERP projects. *Journal of Information Technology*, *15*, 305–316.

Atem de Carvalho, R., & Monnerat, R. (2007). ERP5: designing for maximum adaptability. In A. Oram & G. Wilson (Eds.), *Beautiful code.* O'Reilly.

Atem de Carvalho, R., & Monnerat, R. M. (2008, Sept.–Oct.). Development support tools for enterprise resource planning. *IT Professional*, *10*(5), 39–45.

Bertrand, O., Calonne, A., Choppy, C., Hong, S., Klai, K., Kordon, F., et al. (2009). Verification of large-scale distributed database systems in the NEOPPOD project. In *Workshop on petri nets and software engineering* (pp. 315–316).

Brehm, L., Heinzl, A., & Markus, M. L. (2001). Tailoring ERP systems: A spectrum of choices and their implications. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Maui, Hawaii.* New York, N.Y: Institute of Electrical and Electronics Engineers.

Campos, R. de, Carvalho, R. Atem de, & Rodrigues, J. S. (2007, May). *Enterprise modeling for development processes of open-source ERP.* Paper presetned at the 18th Production and Operation Management Society Conference, Dallas, TX.

Chen, I. (2001). Planning for ERP systems: analysis and future trend. *Business Process Management Journal*, *7*(5), 374–386.

Courteaud, R. (2009, November 24). *ERP5 proxy field.* Nexedi SA. Retrieved July 8, 2010, from `https://www.myerp5.com/kb/document_module/1902`

Davenport, T. H. (1998). Putting the enterprise into the enterprise system. *Harvard Business Review*, *76*(4), 121–131.

Deep, A., Guttridge, P., Dani, S., & Burns, N. (2008). Investigating factors affecting ERP selection in made-to-order SME sector. *Journal of Manufacturing Technology Management*, *19*(4), 430–446.

Foundation for a Free Information Infrastructure. (2009, November). *Total information outsourcing (tio).* Retrieved June 13, 2010, from `http://tio.ffii.org`

Gorny, B., Nowak, Ł., & Perrin, J. (2008, March 28). *How to use property sheets.* Nexedi & ERP5 Community. Retrieved July 7, 2010, from `http://www.erp5.org/HowToUsePropertySheets`

Hevner, A. R., March, S. T., & Park, J. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105.

Hofmann, P. (2008, July–August). ERP is dead, long live ERP. *Internet Computing, IEEE*, *12*(4), 84–88.

Hompel, M. T. (2007). *Materialflusssysteme förder- und lagertechnik mit 36 tabellen* (3., völlig neu bearb. Aufl. ed.). Berlin;Heidelberg;New York: Springer.

Honoré, J. (2009, December 8). *TioLive features summary.* TioLive LLC. Retrieved June 13, 2010, from https://www.tiolive.com/feature

Honoré, J. (2010a, March). *Create report.* Nexedi SA. Retrieved May 11, 2010, from https://www.myerp5.com/kb/document_module/2194?format=pdf

Honoré, J. (2010b, February 24). *OSOE project.* TioLive LLC. Retrieved June 16, 2010, from http://www.osoe-project.org/project

Honoré, J., Robin, S., & Smets, J.-P. (2010, March). *ERP5 products documentation.* Nexedi SA. Retrieved May 11, 2010, from https://www.myerp5.com/kb/web_page_module/1912

Honoré, J., & Smets, J.-P. (2010, February 3). *TioLive partners.* TioLive LLC. Retrieved May 10, 2010, from https://www.tiolive.com/partner

Janssens, G., Kusters, R. J., & Heemstra, F. (2007, June). Clustering ERP implementation project activities: A foundation for project size definition. In S. Sadiq, M. Reichert, K. Schulz, J. Trienekens, C. Moller, & R. J. Kusters (Eds.), *Proceedings of the 1st international joint workshop on Technologies for Collaborative Business Processes and Management of Enterprise Information Systems, Funchal, Portugal* (pp. 23–32). Portugal: Institute for Systems and Technologies of Information.

Kinni, T. B. (1995). Process improvement, part 2. *Industry Week/IW*, *244*(4), p. 45.

Koh, S. C. L., & Simpson, M. (2007). Could enterprise resource planning create a competitive advantage for small businesses? *Benchmarking: An International Journal*, *14*(1), 59–76.

Monnerat, R. M., Carvalho, R. Atem de, & Campos, R. de. (2008). Enterprise systems modeling: the ERP5 development process. In *Proceedings of the 2008 ACM symposium on Applied Computing* (pp. 1062–1068). New York, NY: ACM.

Morabito, V., Pace, S., & Previtali, P. (2005). ERP marketing and Italian SMEs. *European Management Journal*, *23*(5), 590–598.

Nah, F. F. H., Lau, J. L. S., & Kuang, J. (2001). Critical factors for successful implementation of enterprise systems. *Business Process Management Journal*, *7*(3), 285–296.

Nexedi SA. (n.d.). *Nexedi opensource on demand.* Retrieved May 10, 2010, from http://www.nexedi.com

Nexedi SA. (2007, Septmeber 15). *ERP5 Express: World first ERP ASP in open source* [Press release]. Retrieved June 16, 2010, from http://www.erp5.com/news/express-news-release

Nexedi SA. (2009). *TIO-configuration template.* (Restricted)

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). Outline of a design science research process. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, PA.* New York, NY: Association for Computing Machinery.

Perrin, J., & Smets, J.-P. (2010, 02 22). *ERP5 security model overview.* Nexedi SA. Retrieved July 11, 2010, from https://www.myerp5.com/kb/document_module/1546

Rother, K. (2007, December). *ERP5 express user guide.* Nexedi SA. Retrieved May 11, 2010, from https://www.myerp5.com/kb/document_module/289?format=pdf

Russell, S. J., & Norvig, P. (2003). *Artificial intelligence: a modern approach* (2. ed., internat. ed. ed.). Upper Saddle River, NJ: Prentice Hall.

Scheer, A. (1997). *Wirtschaftsinformatik : Referenzmodelle für industrielle geschäftsprozesse* (7., durchges. Aufl. ed.). Berlin: Springer.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1–47.

Shanks, G., & Seddon, P. (2000). Editorial. *Journal of Information Technology (Routledge, Ltd.)*, *15*(4), 243–244.

Smets, J.-P. (n.d.). *ERP5 implementation.* Nexedi SA. Retrieved October 10, 2009, from `https://www.myerp5.com/kb/documentation_section/consultant/consultant-Front.Page/consultant-Implementation.Process/view` (Restricted)

Smets, J.-P. (2004, June). *ERP5: mission-critical erp/crm with python and zope.* Python Software Foundation. Retrieved May 10, 2010, from `http://www.python.org/about/success/nexedi/`

Smets, J.-P. (2007, December). *ERP5 general introduction 2007.* Nexedi SA. Retrieved May 11, 2010, from `https://www.myerp5.com/kb/document_module/158?format=pdf`

Smets, J.-P. (2008, March). *ERP5 industries overview.* Nexedi SA. Retrieved May 11, 2010, from `https://www.myerp5.com/kb/web_page_module/233`

Smets, J.-P. (2009, March 2). *TioLive open source technologies.* TioLive LLC. Retrieved June 13, 2010, from `https://www.tiolive.com/feature/tiolive-technology`

Smets, J.-P., & Honoré, J. (2010, May 5). *OSOE evaluation.* Nexedi SA. Retrieved July 15, 2010, from `https://www.myerp5.com/kb/document_module/2117/view`

Smets-Solanes, J. (2002). *ERP5: a technical introduction.* Paper presented at Linux Tag, Karlsruhe, Germany. Available from `http://cps.erp5.org/sections/free/erp/linuxtag.pdf/view`

Smets-Solanes, J., & Carvalho, R. Atem de. (2003, July–August). ERP5: a next-generation, open-source ERP architecture. *IT Professional*, *5*(4), 38–44.

Snider, B., Da Silveira, G., & Balakrishnan, J. (2009). ERP implementation at SMEs: analysis of five Canadian cases. *International Journal of Operations and Production Management*, *29*(1), 4–29.

Timbrell, G., & Gable, G. (2002). The SAP ecosystem: a knowledge perspective. In *Proceedings of the Information Resources Management Association international conference* (pp. 1115–1118). Hershey, PA: Information Resources Management Association.

TIO Libre Non Profit LLC. (n.d.-a). *TIO definition.* Retrieved June 14, 2010, from `http://www.tiolibre.com/wiki/tiolibre-Wiki.Overview/tiolibre-TIO.Definition`

TIO Libre Non Profit LLC. (n.d.-b). *TIO Libre definition.* Retrieved June 14, 2010, from `http://www.tiolibre.com/guideline/tiolibre-Libre.Definition`

TIO Libre Non Profit LLC. (n.d.-c). *TIO Libre solutions.* Retrieved June 14, 2010, from `http://www.tiolibre.com/solution`

TioLive LLC. (2009, December 11). *TioLive subscription options.* Retrieved June 13, 2010, from `https://www.tiolive.com/service`

TioLive LLC. (2010, April 21). *TioLive grid debut* [Press release]. Retrieved June 15, 2010, from `https://www.tiolive.com/news/news-tiolive-grid-debut`

United Nations Statistics Division. (2008). *International standard industrial classification of all economic activities.* United Nation. Retrieved July 15, 2010, from `http://unstats.un.org/unsd/cr/registry/regdntransfer.asp?f=135`

Wang, L., Tao, J., Kunze, M., Castellanos, A. C., Kramer, D., & Karl, W. (2008). Scientific cloud computing: early definition and experience. In *10th IEEE international conference on High Performance Computing and Communications* (pp. 825–830). New York, N.Y: Institute of Electrical and Electronics Engineers.

Appendix A Content of the Compact Disk

The compact disc which is attached to this thesis contains, the following folders and files:

- `Configuration Case/`: Files of the Aurora Systems configuration case
  - `Aurora-Configuration.*`: The spreadsheet which holds the category configuration of the configuration case.
  - `Aurora-Implementation.*`: The answers to the configuration questionnaire.
- `Diploma Thesis/`
  - `diploma_thesis.pdf`: The diploma thesis in electronic form.
  - `Latex/`: The source files of the diploma thesis.
- `Expert Interviews/Interview Jean-Paul Smets 2009-11-02.ogg`: The expert interview with Jean-Paul Smets, conducted on November 2, 2009.
- `Expert Questionnaires/`: The replies from Jean-Paul Smets and Thierry Brettnacher to the expert questionnaires.
- `Prototypical Implementation/`: The files that make up the created prototypes.
  - `ERP AI Toolkit`: The Zope Product and the ERP5 Business Template that form the ERP5 AI Toolkit
  - `Machine Learning`: The machine learning code examples
    - `tables/`: The tables generated by the binary classification examples.
    - `binary_classification.py`: The script that automates role configuration based on binary classification.
    - `text_classification`: The script that automates product line configuration based on text classification.
  - `Site decsion tree`: The site decision tree as .dia file and as .xml which can be imported into the EAT design tool.
- `References/`: The references, which are available in electronic form.

Appendix B Output of the Text Classification Example

```
############################
Fish plus
############################
Description:  Fish plus company purchase fish, then cuts it and
distributes to end users through local shops in the Lille area. All fish
are wrapped in paper or boxes. The company is managed by fishermen who
rely on external accountant.

Original: [’fish’, ’packaging’, ’service’]
packaging [(’y’, 0.95716241309377637), (’n’, 0.23653124484777444)]
fish [(’y’, 0.7453542438870302), (’n’, 0.73150671283995972)]
cafe [(’n’, 0.93143565146697616), (’y’, 0.27965628443071877)]
computer [(’n’, 0.95716241309377637), (’y’, 0.23653124484777444)]
service [(’y’, 0.99990000000000001)]
############################
Bocafé
############################
Description:  Bocafé imports café from Brazil and sells
it in France through an ecommerce shop. All sales are done online.
Imported café is biologic. Accunting is handled by external accoutant.
Bocafé has about 10 employees, three café experts and 7 e-commerce
cracks.

Original: [’cafe’, ’packaging’, ’service’]
packaging [(’y’, 0.9582454419108466), (’n’, 0.23653124484777444)]
fish [(’n’, 0.93358071970455248), (’y’, 0.25636355885841361)]
cafe [(’y’, 0.77161378151482807), (’n’, 0.70656933794879184)]
computer [(’n’, 0.9582454419108466), (’y’, 0.23653124484777444)]
service [(’y’, 0.99990000000000001)]
############################
Nexedi
############################
Description:  Nexedi develops custom made business application based
on ERP5. All services are done internally. ERP5 is a Zope-based ERP
system. It’s also available as a free Software as a Service (SaaS).
Nexedi is located in Lille, France

Original: [’service’, ’computer’]
```

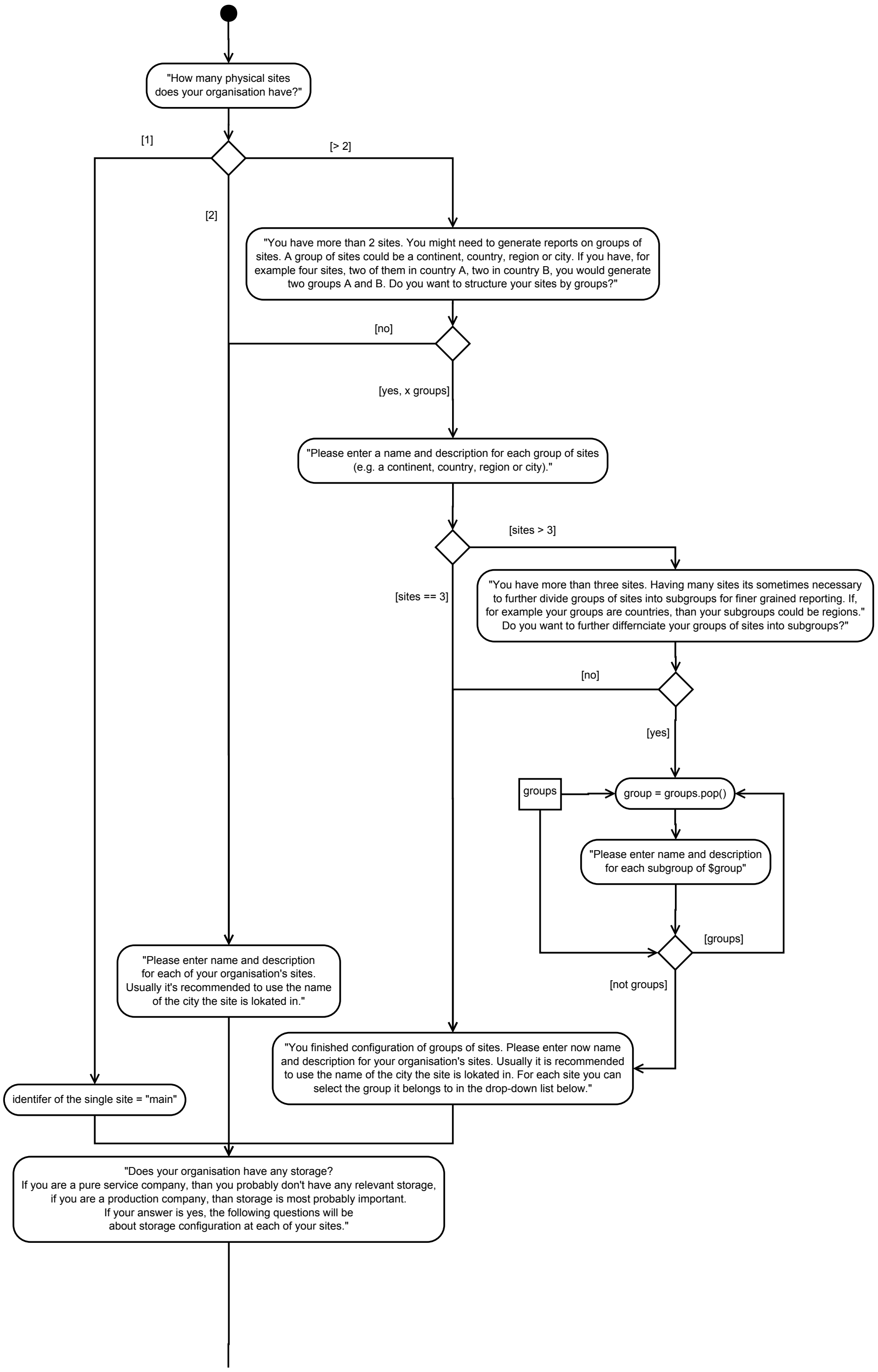
```
packaging [(’n’, 0.77223690129709766), (’y’, 0.74170355919588682)]
fish [(’n’, 0.95006922234888924), (’y’, 0.24433802261746562)]
cafe [(’n’, 0.94777833630938024), (’y’, 0.25672993919827664)]
computer [(’y’, 0.77223690129709766), (’n’, 0.74170355919588682)]
service [(’y’, 0.99990000000000001)]
############################
Omega software
############################
Description:  Omega software sells library applications to schools
in Germany. Everything is done internally.

packaging [(’y’, 0.86738278128548574), (’n’, 0.57578875488183334)]
fish [(’n’, 0.91188836044136834), (’y’, 0.58801795902970122)]
cafe [(’n’, 0.85690973029582596), (’y’, 0.58395477344197833)]
computer [(’n’, 0.86738278128548574), (’y’, 0.57578875488183334)]
service [(’y’, 0.99990000000000001)]
```

Appendix C Output of the Binary Classification Example

```
###############
Nexedi
###############
Answers   : {’client-types’: (’business’, ’administration’,
’not-for-profit’), ’sell-what’: (’services’,), ’sell-how’:
(’online’, ’distributor’, ’other’), ’not-for-profit’: ’n’}
Original  : [’role/admin’, ’role/client’, ’role/client/dis
tributor’, ’role/internal’, ’role/media’, ’role/saleslead’,
’role/salesprospect’, ’role/supplier’, ’role/user’]
Classified: [’role/admin’, ’role/client’, ’role/client/dist
ributor’, ’role/internal’, ’role/media’, ’role/saleslead’,
’role/salesprospect’, ’role/supplier’, ’role/user’]
###############
Fish shop
###############
Answers   : {’client-types’: (’consumer’,), ’sell-what’:(’p
roducts’,), ’sell-how’: (’retail’,), ’not-for-profit’: ’n’}
Original  : [’role/admin’, ’role/client’, ’role/internal’,
’role/supplier’]
Classified: [’role/admin’, ’role/client’, ’role/internal’,
’role/supplier’]
###############
Rotary Club
###############
Answers   : {’client-types’: (), ’sell-what’: (), ’sell-how
’: (), ’not-for-profit’: ’y’}
Original  : [’role/admin’, ’role/client’, ’role/member’, ’r
ole/supplier’]
Classified: [’role/admin’, ’role/client’, ’role/member’, ’r
ole/supplier’]
###############
Aurora Systems
###############
Answers   : {’client-types’: (’not-for-profit’,), ’sell-wha
t’: (’products’,), ’sell-how’: (’online’, ’other’), ’not-fo
r-profit’: ’n’}
Original  : [’role/admin’, ’role/client’, ’role/internal’,
’role/saleslead’, ’role/salesprospect’, ’role/supplier’,’role/user’]
```

```
Classified: [’role/admin’, ’role/client’, ’role/internal’,
’role/saleslead’, ’role/salesprospect’, ’role/supplier’, ’role/user’]
################
Some not-for profit organization
################
Answers   : {’client-types’: (), ’sell-what’: (), ’sell-how’
: (), ’not-for-profit’: ’y’}
Classified: [’role/admin’, ’role/member’,]
################
Some company
################
Answers   : {’client-types’: (’business’, ’not-for-profit’)
, ’sell-what’: (’products’,), ’sell-how’: (’online’,), ’not
-for-profit’: ’n’}
Classified: [’role/admin’, ’role/client’, ’role/internal’,
’role/saleslead’, ’role/salesprospect’, ’role/supplier’, ’role/user’]
```

Appendix D Draft Version of the Site Decision Tree



"How many physical sites does your organisation have?"

"You have more than 2 sites. You might need to generate reports on groups of sites. A group of sites could be a continent, country, region or city. If you have, for example four sites, two of them in country A, two in country B, you would generate two groups A and B. Do you want to structure your sites by groups?"

"Please enter a name and description for each group of sites (e.g. a continent, country, region or city)."

"You have more than three sites. Having many sites its sometimes necessary to further divide groups of sites into subgroups for finer grained reporting. If, for example your groups are countries, than your subgroups could be regions." Do you want to further differnciate your groups of sites into subgroups?"

group = groups.pop()

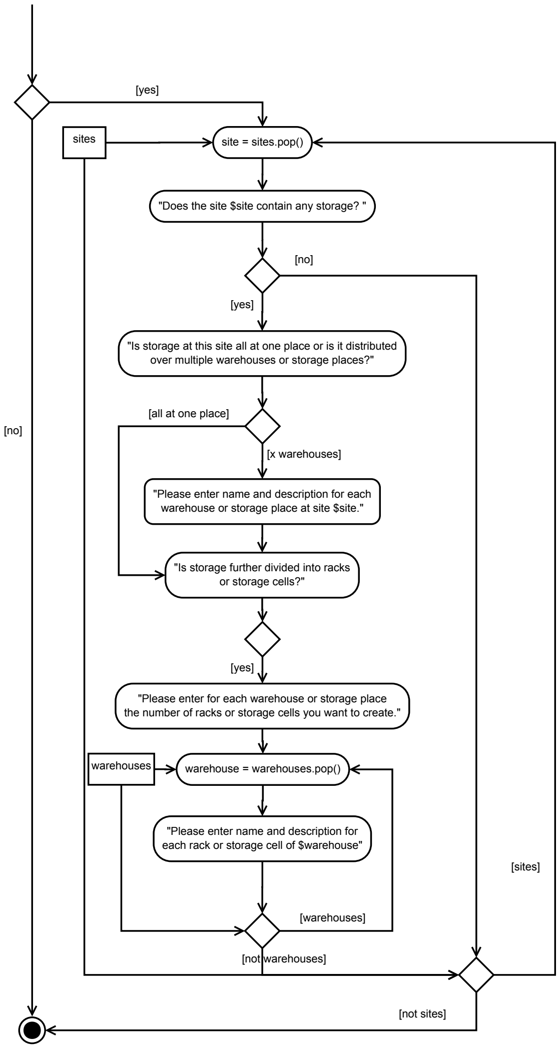
"Please enter name and description for each subgroup of $group"

"Please enter name and description for each of your organisation's sites. Usually it's recommended to use the name of the city the site is lokated in."

"You finished configuration of groups of sites. Please enter now name and description for your organisation's sites. Usually it is recommended to use the name of the city the site is lokated in. For each site you can select the group it belongs to in the drop-down list below."

identifer of the single site = "main"

"Does your organisation have any storage?
If you are a pure service company, than you probably don't have any relevant storage,
if you are a production company, than storage is most probably important.
If your answer is yes, the following questions will be
about storage configuration at each of your sites."


[yes]
sites
site = sites.pop()
"Does the site $site contain any storage? "
[no]
[yes]
"Is storage at this site all at one place or is it distributed over multiple warehouses or storage places?"
[all at one place]
[no]
[x warehouses]
"Please enter name and description for each warehouse or storage place at site $site."
"Is storage further divided into racks or storage cells?"
[yes]
"Please enter for each warehouse or storage place the number of racks or storage cells you want to create."
warehouses
warehouse = warehouses.pop()
"Please enter name and description for each rack or storage cell of $warehouse"
[sites]
[warehouses]
[not warehouses]
[not sites]

Appendix E Categories of the Configuration Case

Site

Path	ID	Title	Short Title	Description
*	hoechberg	Höchberg	Höchberg	Office in Höchberg
*	dresden	Dresden	Dresden	Office in Dresden

Group

Path				ID	Title	Short Title	Description
*				aurorasystems	Aurora Systems	Aurora	Aurora Systems is a small IT company specialised in the development of administration software For school libraries
	*			sales	Sales and Distribution	Sales	Sales department is in charge of distribution, marketing, sales as well as administrative work
*				germany	Administrations in Germany	Germany	German administration, governmental organisations and public insitutes
	*			bavaria	Administrations in Bavaria	Bavaria	Bavarian administration, governmental organisations and public insitutes
		*		education	Bavarian ministry of education and science	Education	The Bavarian ministry of education and science
			*	bsb	Bayerische Staatsbibliothek	BSB	The Bavarian state library has various satallite stations in Bavaria (Nürnberg, Würzburg, München, Regensburg) which advise schools in their region
			*	isb	Institut für Bildungsforschung	ISB	The Institut für Bildungsforschung is divided into various departements, some of which advise Schools about library software
		*		würzburg	Administrations in Würzburg	Würzburg	Würzburg administration organisations
		*		munich	Administrations in Munich	Munich	Munich administration organisations

Function

Path	ID	Title	Short Title	Description
*	sales	Sales and Distribution	Sales	An entity in charge of Sales and Distribution
*	agent	Sales Agent	Agent	A person in charge of executing sale orders, packaging and distribution
*	marketing	Marketing	Marketing	An entity in charge of Marketing
*	agent	Marketing Agent	Agent	A person in charge of designing marketing material like flyers etc.
*	company	Company	Company	A company is a legal entity which has been registered at a commerce registry and which Has full autonomy.
*	executive	Company Executive	Executive	A company executive has broad decision power and broad access to confidential Information of the company.
*	agent	Company Agent	Agent	A company agent is a regular staff of the company in charge of operations. He or she has Little or no decision power.
*	education	Education	Education	An educational organisation such as a school
*	manager	Education Manager	Manager	A manager in an educational organisation such as a school director
*	agent	Education Agent	Agent	An agent in an educational organisation such as a school secretary
*	professor	Professor	Professor	A professor in an educational organisation
*	admin	Public Administration and institutes	Administration	Public administration and istitutes like the education departement of a city, the bavarian Institute of the bavarian state library
*	manager	Public Administration Manager	Manager	A public administration manager has decision power
*	agent	Public Administraton Agent	Agent	An public administration agent is regular staff of the public administration in charge of Operations

Role

Path	ID	Title	Short Title	Description
*	multiplier	Multiplier	Multiplier	A multiplier is somone who influences the opinion of our prospects and leads. For example someone part of an organisation Which advises schools about library products
*	internal	Staff	Staff	Corporate staff
*	client	Client	Client	Client
*	supplier	Supplier	Supplier	Supplier
*	admin	Administration	Administration	Public administration, tax office
*	user	User	User	A registered user of meine-schulbibliothek.de
*	lead	Sales Lead	Lead	A person or an organisation that is potentially interested in purchasing a product or service from our compagny group (first stage In sales process)
*	prospect	Sales Prospect	Prospect	A person or an organisation that is interested in purchasing a product or service from our compagny group (qualified sales lead)

Grade

Path	ID	Title	Short Title	Description
*	employee	Employee	Employee	Full time or part time employee
*	trainee	Trainee	Trainee	Trainee
*	associate	Associate	Associate	Associate or owner of the company

Region

Path			ID	Title	Short Title	Description
*			germany	Germany	Germany	Germany is our main marked. The most important distinction is between differen Bundesländer Beacause there are different school laws and schools adminstrations in each Bundesland
	*		bavaria	Bavaria	Bavaria	Most of our schools are located in Bavaria, so we further distinguish between different Regierungsbezirke as its important to our clients to know how many other Kallimachos Schools are located in ther nearest region
		*	unterfranken	Unterfranken	Unterfranken	Schools in Unterfranken
		*	oberfranken	Oberfranken	Oberfranken	Schools in Oberfranken
		*	mittelfranken	Mittelfranken	Mittelfranken	Schools in Mittelfranken
		*	oberpfalz	Oberpfalz	Oberpfalz	Schools in Oberpfalz
		*	schwaben	Schwaben	Schwaben	Schools in Schwaben
		*	oberbayern	Oberbayern	Oberbayern	Schools in Oberbayern
		*	niederbayern	Niederbayern	Niederbayern	Schools in Niederbayern
		*	munich	München	München	Schools in Munic (Also Munic is a city, it's a Regierungsbezirk of its own)
	*		baden-wuerttemberg	Baden-Württemberg	Baden-Württemberg	Schools in the Bundesland Baden-Württemberg
	*		hessen	Hessen	Hessen	Schools in the Bundesland Hessen
	*		niedersachsen	Niedersachsen	Niedersachsen	Schools in the Bundesland Niedersachsen
	*		berlin	Berlin	Berlin	Schools in the Bundesland Berlin
*			luxemburg	Luxemburg	Luxemburg	Interested schools in Luxemburg
*			suisse	Switzerland	Switzerland	Interested schools in Switzerland

Skill

Path				ID	Title	Short Title	Description
*				management	Management		Management skills
	*			marketing	Marketing	Marketing	Ability to create marketing materials and marketing concepts
	*			communication	Communication	Communication	For advertising
*				sales	Sales and Distribution	Sales	Skills related to sales an distribution
	*			advise	Advise clients	Advise	Ability to advise a client about what functionality he needs in the event of selling a Kallimachos license or meine-schulbibliothek.de contract
	*			distribution	Distribution	Distribution	Skills related to distribution and packaging of our products
		*		configuration	Kallimachos Configuration	Configuration	Abilty to confgure the Kallimachos library Program according to a sale order
		*		scanner	Adjusting Scanner	Scanner	Ability to correctly adjust a barcode scanner for sale
		*		barcode	Printing Barcode Labels	Barcode	Ability to use the lable printer and software to print custom barcode labels for our clients
*				support	User Support	Suport	Ability to support Kallimachos and meine-schulbibliothek.de users
*				design	Design	Design	Multimedia design skills
	*			graphic	Graphic Design	Graphic	Ability to design graphics, logos, layouts, etc.
*				admin	Administration	Administration	Business administration skills
	*			accounting	Accounting	Accounting	Ability to book keep accounting transactions
*				it	Information Technology	IT	Information Technology skills
	*			sysadmin	System Administration	Sysadmin	System administration skills
		*		linux	Linux System Administration	Linux	Ability to administer a Linux System
		*		macos	MacOS System Administration	MacOS	Ability to administer a MacOS System
		*		windows	Windows System Administration	Windows	Ability to administer a Windows System
	*			webadmin	Website Administration	Website	Abilityto administer a website
	*			programming	Programming	Programming	Skills related to programming
		*		lang	Programming Languages	Languages	Knowledge of programming languages
			*	python	Python	Python	Ability to programme in Python
		*		system	System Programming	System	Ability to programmatically interface with an OS
			*	linux	Linux System Programming	Linux	Ability to programmatically interface with a linux system
			*	macos	MacOS System Programming	MacOS	Ability to programmatically interface with a MacOS system
			*	windows	Windows System Programming	Windows	Ability to programmatically interface with a Windows system
		*		application	Application Programming	Application	Application programming skills
		*		web	Web Application Programming	Web	Web application programming skills
*				lang	Language	Language	Ability to work in a given language
	*			en	English	English	Ability to work in English
	*			de	German	German	Ability to work in German
*				soft	Softskills	Softskills	Skills which are not really technical and not really taught at university
	*			presentation	Presentations	Presentations	Ability to create stunning presentations
	*			speech	Give Presentation	Speech	Abiltiy to give convincing presentations
	*			formatting	Document Formating	Formatting	Ability to create well formatted documents
*				library	Library Management	Library	Librarian and library management skills

Activity

Path	ID	Title	Short Title	Description
*	education	Education	Education	Education
*	school	School	School	Any kind of non-academic school (primary, highschool, professional school)
*	university	University	University	Universities and higher education (ex. Kyoto University, Lille3, Telecom Lille1, Université de Dakar, MIT)
*	industry	Industry	Industry	Industry and services
*	banking	Banking and Finance	Banking	Banking and finance (ex. HSBC, BNP Paribas, Citybank)
*	logistic	Logistic	Logistic	Logistics (ex. UPS, CGM)
*	it	Information Technology	IT	Information technology (ex. IBM, Bull, NEC)
*	expendable	Expendable items	Expendable	Expendable items used in conjunction wit it, like barcode labels
*	hardware	Hardware	Hardware	Provider of hardware such as servers, laptops, routers (ex. Asus, Dell, Supermicro)
*	consulting	IT Consulting	Consulting	Provider of consulting services for network, applications, security (ex. IBM Global Services)
*	webagency	Web Agency	Web Agency	Provider of web site design and implementation (ex. Quadra)
*	software	Software (Proprietary)	Software	Proprietary software publisher (ex. Microsoft)
*	floss	Open Source - Free Software	FLOSS	Open Source software publisher and service provider
*	nonprofit	Non-profit	Non-profit	Non-profit organisations (NGO)
*	association	Non-profit Association	Association	Not for profit associations (ex. April, AFUL, MSF)
*	foundation	Non-profit Foundation	Foundation	Foundations (ex. FSF, Fondation de France)
*	government	Government	Government	Government organisations
*	town	City	City	Towns and cities (ex. Paris City, Tokyo City, Dakar City, Campos City)
*	regional	Regional Government	Regional Government	Regional government (ex. California Government, Bavaria Länder, Catalonia Generalitat, Lorrain Region)
*	agency	Regional Agency	Agency	For example regional eductaion agencies
*	professional	Professional Organisation	Professional	Professional organisations (for profit or not for profit)
*	chamber	Chamber of Commerce	Chamber	Chambers of commerce, business registries

Publication Section

Path		ID	Title	Short Title	Description
*		marketing	Marketing	Marketing	Marketing documents including web site, presentations, leaflets
	*	kallimachos	Kallimachos	Kallimachos	Marketing Documents for Kallimachos
	*	msbde	meine-schulbibliothek.de	msbde	Marketing Documents for meine-schulbibliothek.de
*		news	Global News Feed	News	News about related to marketing, projects, operations, etc. It can be anything.
*		documentation	Documentation	Documentation	Kallimachos documentation for users and Aurora Systems staff.
*		process	Business Processes	Process	Documents related to core business processes
*		technology	Technology Library	Technologies	A collection of articles and links to technologies commonly used Aurora Systems or implemented In Kallimachos
*		template	Template	Template	Document templates of general use (ex. Offer, Letter, Fax.)
*		tax	Tax	Tax	Tax and social declarations
*		other	Other documents	Other	A place holder for documents which can not be categorised with the current categorisation. Based On the content of Other, new categories may be created to improve the categorisation system.

Product Line

Path			ID	Title	Short Title	Description
*			kallimachos	Kallimachos	Kallimachos	Kallimachos library program licenses and services
	*		license	Licenses	License	Kallimachos Licenses with different kinds of functionality
	*		update	Update Licences	Update	Kallimachos update licenses
	*		support	Support Services	Support	Kallimachos support services
	*		programming	Custom Functions Programming	Programming	Kallimachos custom function programming services
*			msbde	meine-schulbibliothek.de	MSBDE	meine-schulbibliothek.de services
	*		hosting	Hosting Services	Hosting	meine-schulbibliothek.de hosting services
	*		support	Support Servies	Support	meine-schulbibliothek.de support services
	*		programming	Programming Services	Programming	meine-schulbibliothek.de custom function programming services
*			expendable	Expendable items	Expendable	Expendable items for library administration
	*		barcode	Barcode labels	Barcode	Barcode labels for books
	*		ink_ribbon	TT Ink Ribbon	Ink Ribbon	Thermo transfer ink ribbon to print barcode labels
	*		film	Protective film	Film	Protective film for barcode labels
*			hardware	Hardware	Hardware	Any hardware
	*		scanner	Barcode Scanners	Scanners	Barcode scanners for libraries
		*	ccd	CCD Scanners	CCD	Cheap CCD scanners
		*	laser	Laser Scanners	Laser	High quality laser scanners

Honorable Declaration

I hereby declare that I have prepared the diploma thesis

> *Automating ERP Package Configuration for*
> *Small Businesses*

myself. Material and ideas directly or indirectly adopted from external sources are properly indicated.

The paper has not been previously presented to an examination board and has also not been published.

Dresden, July 21, 2010