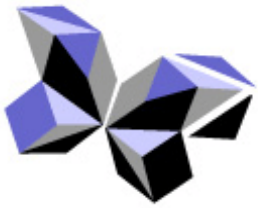


Digital Business Ecosystem

Contract n° 507953

WP 7: User Profiling

Del 7.1: Description of necessary information about DBE customer, to support a long term evolutionary business relationship

Project funded by the European Community under the "Information Society Technology" Programme

Contract Number: 507953

Project Acronym: DBE

Title: Digital Business Ecosystem

Deliverable N°: 7.1

Due date: 30/04/2005

Delivery Date: 20/05/2005

Short Description:

This Deliverable introduces basic ideas on User Profiling, its possible mechanisms and the internal data organisation using a basic BML extension. Additionally a set of possible dimensions and metrics will be provided.

Partners owning: FZI (Christian Bartsch)

Partners contributed: STU, ISUFI, T6, TUC, WIT

Made available to: DBE Partners and European Commission

VERSIONING

VERSION	DATE	AUTHOR, ORGANISATION
0.1	**30/01/2005**	**CHRISTIAN BARTSCH (FZI)**
1.0	**28/02/2005**	**CHRISTIAN BARTSCH (FZI)**
1.1	**01/04/2005**	**CHRISTIAN BARTSCH (FZI)**
1.2	**12/05/2005**	**CHRISTIAN BARTSCH (FZI) + THOMAS KURZ (STU)**
1.3	**16/05/2005**	**CHRISTIAN BARTSCH (FZI) + MAURIZIO DE TOMMASI (ISUFI)**
1.4	**20/05/2005**	**CHRISTIAN BARTSCH (FZI)**

Quality check

1st Internal Reviewer: Thomas Kurz (STU)

2nd Internal Reviewer: Maurizio De Tommasi (ISUFI)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The DBE Project, being an integrated project, has to develop a complex and innovative platform enabling e-business for SMEs, but also strategies and policies enabling the diffusion of a DBE platform in Europe and world wide. The long term objective of the digital ecosystem community is to foster local economic growth through new forms of dynamic business interactions and global co-operation among organisations, territories and business communities enabled by digital ecosystem technologies. The DBE long term strategy stands on the capability of the DBE Project Partners to mobilise regions, territories and communities on its innovative vision. Regions, territories and distributed communities will adopt DBE vision and frameworks creating in the first stage isolated communities that simply move from territory based relations to virtual one. The following step, when the number of regions, territories and communities is large enough and a critical mass is reached, it is based on the emergence of regional nodes interacting in virtual mode with other territories and at the same time on the emergence of virtual companies using the DBE infrastructure to create business interaction among different territories and communities. The last step is the creation of a complex network of business relations that virtually connect every DBE territories, communities and companies as exemplarily shown in Figure 1. Such approach needs, as propulsive energy source the regions and communities capability of innovation which should be enabled and supported by correct policies.

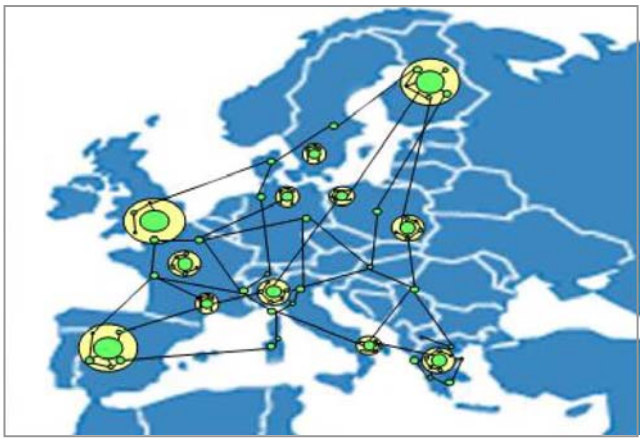


Figure 1: Possible DBE Network in Europe [ISUF04, p.10]

As an important basis for a long-term business relationship between two or more actors a certain level of information about potential business partners is necessary. In this context information specifically means any kind of data regarding a partner including skills, needs, requirement, preferences, offered and demanded services and so on. With the support of this knowledge the possibilities to engage a new business relationship or to intensify

already existing relationships exemplary based on equal preferences or services can increase. One can imagine that isolated information about a company or a user is supposed to be useless in a system of many participants as only a matching of information between two or more Users can add value to a relationship. This happens when at least two actors agree on the benefits of information provided to each other. Especially if more than two actors are involved, this stage could also be seen as the very basic beginning of building a relationship network. The DBE knows four main actors as shown in Figure 2.

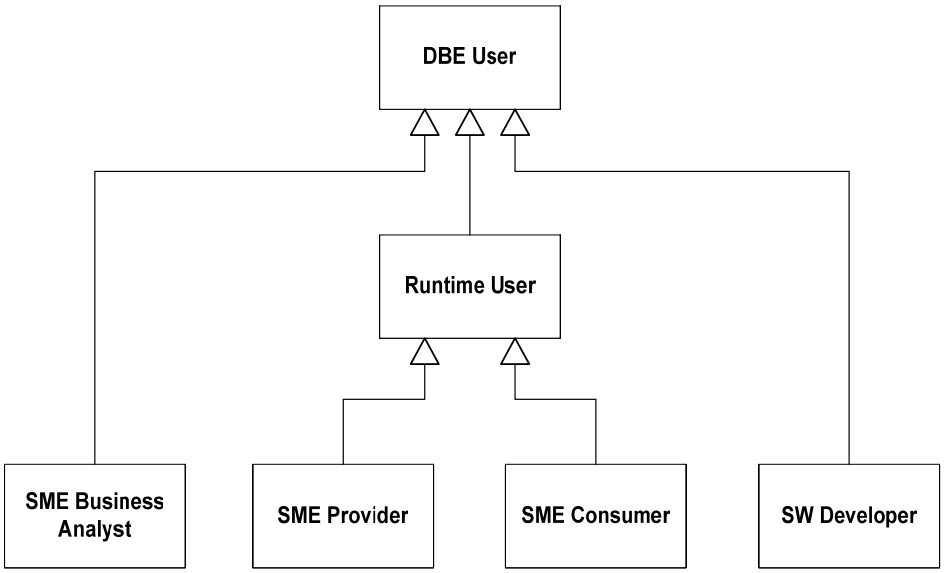


Figure 2: Actors in the DBE [Solu04, p.10]

All classical roles are DBE Users whereas the SME Provider and the SME Consumer are Runtime Users. They can offer and demand Services only at runtime using the DBE System. In contrast to this the SME Business Analyst and the Software Developer usually acts without explicitly using the DBE at runtime. The SME Business Analyst is responsible for collecting and aggregating needs and requirements of similar SMEs within a region in order to develop customised BML Models. A SME Provider makes one or more software services available to the DBE whereas the SME Consumer will use a specific DBE service implementation creating an individual B2B relationship (e-business). The SW Developer is responsible for developing software solutions compliant with specific DBE implementation.
Of course role definition boundaries can become indistinct by-and-by. One imagines the case that the role SME Provider merges with SW Developer as a provider decides not to resell software anymore than to develop software on its own. It is also conceivable for a SME Consumer to switch roles and to become a SME Provider because he wants to start a new software development business using the DBE.

But where does information about a user come from, what information is useful, who maintains it and who takes advantages of this knowledge? The answers to these questions will be part of this Deliverable. As the objective of this paper is focused on the

description of necessary information to provide a long-term business relationship, the detailed mechanisms on how to collect and proceed with user related information will be part of upcomming Deliverables from Work Package 7.
Chapter 2 describes what a DBE business relationship could be about and how the digital environment has influence on that. The DBE User Information in Chapter 3 includes what information of a user is needed to be stored within a profile to provide an adequate mapping in the further course of the project. There will also be a short introduction to the information space of the DBE defined as 'opportunity space' to give an impression of the complexity of possible relations and the almost unlimited number of sources for retrieving information. The core of Chapter 4 will be the basic design and concept of the User Profiling and its related areas of work including a first version of an extended BML for User Profiling purposes.

2 BUSINESS RELATIONSHIP APPROACH

The following chapter discusses the importance of business relationship intentions between DBE network participants in order to generate lucrative and successful business relation networks.
Especially within online business - and the Digital Business Ecosystem is a prime example of a networking online business - a lot of business activities are handled through computers and most business transactions are proceeded via a website, email or internet protocol based telephony. For this reason it is not astonishing that due to the majority of online business transactions the development of a business relationships network by meeting people one-on-one appears quite difficult. Personal conversation opportunities turn out to be more unfrequent, therefore, as with any online business, the main focus on building these relationships lies on the internet.
Although this may be rather more difficult compared to meeting a customer in person, a sufficient business relationship can still be accomplished by keeping communications open and establishing trust with the customers, as it is an essential step to business success. Regarding this issue an often made mistake is starting an online business only focusing on what services or products can be sold. There is little concern about establishing a business relationship or even a simple "thank you" after a successful business transaction. Using this methodology, suppliers might see a significant drop in their business over time as customers won't deal with a company that gave poor customer service [Kova05].
Thinking of relationship building as the foundation to a business, a company could establish itself as a professional, trust worthy and a reliable source, leaving a positive

impression to all doing business with it. This aspect becomes significantly more important when mentioning the possible internet based delivery channels like [Dych02]:

- 24-hour-access
  (access to the internet at anytime and anywhere)
- up-to-the-minute information
  (services like lookup for product features and prices, stock level information, …)
- online customer support
  (helpdesk, online support requests, real-time FAQ handling, …)
- online self service
  (lookup for manuals, download possibilities, search engines, …)
- personalised content
  (using user profiles generated from explicit User knowledge and implicit User behaviour)

Almost every business partner has the possibility to use aforementioned channels as intensive as he likes to. Therefore a long-term relationship between business partners calls for a very good knowledge on what the business partners needs and requirements are and how he evolves during a business partnership. With this background it appears reasonable to focus on the importance of an intelligent user profiling architecture providing chances for collecting and maintaining useful user data in order to represent the real world user profile as detailed as possible within the DBE. By doing this all delivery channels can be taken into account and additionally DBE users, no matter whether it is a Consumer or a Provider, will be able to request and provide, sell and buy, develop and consume services on the way they prefer. But not only the delivery channel will be the most suitable one, also suggested or recommended products or services can be used for matching the SMEs preferences. An introduction to the concept and mechanisms for achieving this high-level objective is available from Chapter 4 on. As it is the core of the next Deliverable D7.2 a more detailed description of the user Profiling mechanisms will be given there. Of course many further factors exist having influence on a long-term business relationship approach just like economy, politics and governmental organisations, environmental conditions, legal aspects and so on. From the User Profiling point of view the most important and within the bounds of this work package feasible task is the design and development of a User Profiling which supports the long-term relationship idea as efficient as possible.

3 DBE USER REQUIREMENTS

3.1 RELATED DBE COMPONENTS

Before describing DBE user requirements for user profiling purposes (see Chapter 4) it is necessary to identify which other components provide and request information to and from a user profile later on. Among others this is important because it generally makes clear what DBE components work together in a direct or indirect way. It is also possible to identify potential resources for collecting user related information. This should also be taken into account when designing user profiling interfaces on the one hand and implementing the user profiling mechanisms on the other hand. The following mentioned DBE components are already affiliated to the user profile or will be affiliated to it in the further course of the project. They are shortly described in reference to their connection to the user profiling as such:

The Recommender System (RS)

The Recommender component in the DBE is used to personalise the DBE knowledge for each user (end user or SME). In addition to the classical recommendation system function '*matching user profiles with existing knowledge and making autonomously recommendations to the user in a push mode*' this component will also be used in two different modes:

- **Search Mode**
  The user makes an explicit request to the Recommender in order to access knowledge that satisfies some criteria. This is a classical Information Retrieval (IR) scenario where some person (or entity) engages in an active information-seeking behaviour, such as submitting a query to an IR system. This mode is also called "*pull mode*".

- **Recommendation Mode**
  The Recommender component acts as an autonomous service (or agent) on behalf of the user and tries to filter the DBE knowledge and make valuable recommendations to the user. In order to make valuable recommendations, the

Recommender needs to know the user's preferences and requirements. This information will be provided by the user profile.

Conceptually, for the Recommender a user profile should contain the description of the desired knowledge in the same way that this knowledge has been described for recommendation issues. A user profile should also describe the services and/or the partners that a user wants to find, possibly enriched with Fuzzy operators (for more details see D17.1). Then the Recommender will try to match these descriptions with underlying information of the DBE knowledge base (either in the Service Factory Environment - SFE or in the Execution Environment - ExE). For that the existence of a MOF (Meta Object Facility) user profile meta-model is favoured as in such case the user profiles will contain instances (models) of this meta-model. From the Recommender point of view it would be useful, if the user profile comprises a set of preferences (standing queries) on specific segments of knowledge (e.g. Business Models, Semantic Service Descriptions, etc.). Regarding the technical aspect, the recommender has already defined a Query Modelling Language (QML) that can express user requests (preferences) for all kinds of knowledge which is captured exemplary with BML, SSL, ODM, SDL in a uniform way. Thus, this Language could possibly act as the basis that will be used by the profiling mechanism to internally represent the user preferences. How this idea can be adapted by the User Profiling will be part of the further work on WP7. With this approach, the profiling mechanism could also exploit the knowledge base for its persistency requirements. As the technical integration and interconnection of the user profiling mechanism and the Recommender will be part of the next Deliverable D7.2 (Initial Description of Profiling Mechanism Design) further implementation aspects will be discussed there.

The Evolutionary Environment (EvE)

Work package 9 deals with the concept of fitness as a fundamental driving force in evolution which will be delivered to the Digital Business Ecosystem by including biology to it. The design of the EvE is based on the claimed open infrastructure that is capable of supporting spontaneous evolution, composition and adaptation of software components and services. The connection to the user profiling arises from the fitness functions need, to be additionally fed with user information in order to work properly. From the EvE point of view the user profiling should function as an individual user repository which stores preferences regarding requests and services for optimised matching purposes. In the course of time the user profile is also expected to provide a more detailed picture of a user

and his preferences using the DBE by requesting and choosing services. The user profiling is also a good starting point for pre-configuration. Parameters derived from manual edit of user preferences and general settings can initially be delivered to the EvE's fitness function for the optimisation process. Especially preferences like *"companies > 500 employees ; provider distance = local ; Web Service Availability = high"* and so on seem to be useful for the EvE as they can easily be used for searching purposes. Therefore the information focus is mainly on business preferences as they are closely related to the fitness function optimisation (see also Deliverable D9.1). Overall it should be kept in mind that the EvE works starting from a CIM user request expressed in SBVR through the BML editor.

Accounting

Work package 36 delivers a set of Accounting software building blocks. A detailed description can be found in [WIT04]. More specific WP36 will provide accounting (metering) as an infrastructural service and accounting mediation as DBE services. Basically the key to the accounting process will be the user identity. As the accounting process uses data which seems to be useful to be part of the user profile following information should at least be stored within it:

- Billing Address
- Delivery Address
- Billing Date
- Preferred method of Payment (Credit Card, Terms of Payment etc.)
- Preferred method of contact (Communication Channels: Email, Phone, Fax etc.)

From the Accounting point of view a very interesting and useful feature would be the storage of data and metrics derived from accounting data could be as follows: Accounting data could be analysed to create a metric indicating if a user has a preference for high quality charged services or lower quality/value free services. Of course an indication will be necessary if a DBE user (service provider) is a provider of accounted (pay for) service.

3.2 THE DBE OPPORTUNITY SPACE

In the beginning of the project the expression 'target domains' was introduced to provide real life business focus and environment for the DBE project and system. The initial

collection and consolidation of suitable user information adapted for the DBE depends on choosing target domains which deliver adequate information in order to use this knowledge for further generalisation. Although a target domain in this context could at the first view be interpreted as an industry sector this allocation seems to be too restrictive for aforementioned purposes. Mentioning only a simple industry sector would probably not lead to a representative mapping of the real world because inter-organisational dependencies and cross-organisational networking activities would not be taken into account so deeply [TDM04, p.2]. As a result of this an extended understanding of a target domain is necessary. This leads to the DBE Opportunity Space which additionally tries to consider organisational networking aspects to deliver more detailed information on how SMEs are currently connected with each other starting just from one single centred entity. According to the projects regional catalysts an example of how an opportunity space could look like is presented in Figure 3. As this should just be an introduction to the expression 'Opportunity Space' the level of this description is kept high to generally explain from what business sources information for a user profile could be retrieved. If one goes into deeper levels of the opportunity space's visualisation and description following components should also be contained within it [TDM04, p.5]:

- a model of actor (company or public body) types and their relationships
- strategic driving forces in these relationships
- typical / critical business transactions performed in the relationships
- data model of data which is temporarily or permanently stored
- value added by the DBE infrastructure (business view)
- software view of the DBE infrastructure services which can be used to support the business scenarios

Although the idea of a n-dimensional space including the definition of multiple horizontal dimensions would deliver a more detailed view on an opportunity space and its connections it is assumed not to be useful because of the limited number of SME partners participating this project. A representative evaluation of an opportunity space framework might become more relevant with an increasing SME population. But also the level of complexity for such a model would probably increase dramatically. Instead, in the first iteration a prototype approach was chosen to clarify the basic layout of an opportunity space including possible opportunities for B2B (and B2C) applications.
Having a closer look on the opportunity space 'Tourism' following example, based on interviews and use cases gained from Task B4, will give a better understanding of what an opportunity space could be about. Following graphic exemplarily illustrates an opportunity space starting from the entity basis 'Hotel' which is a well-defined type of SME.

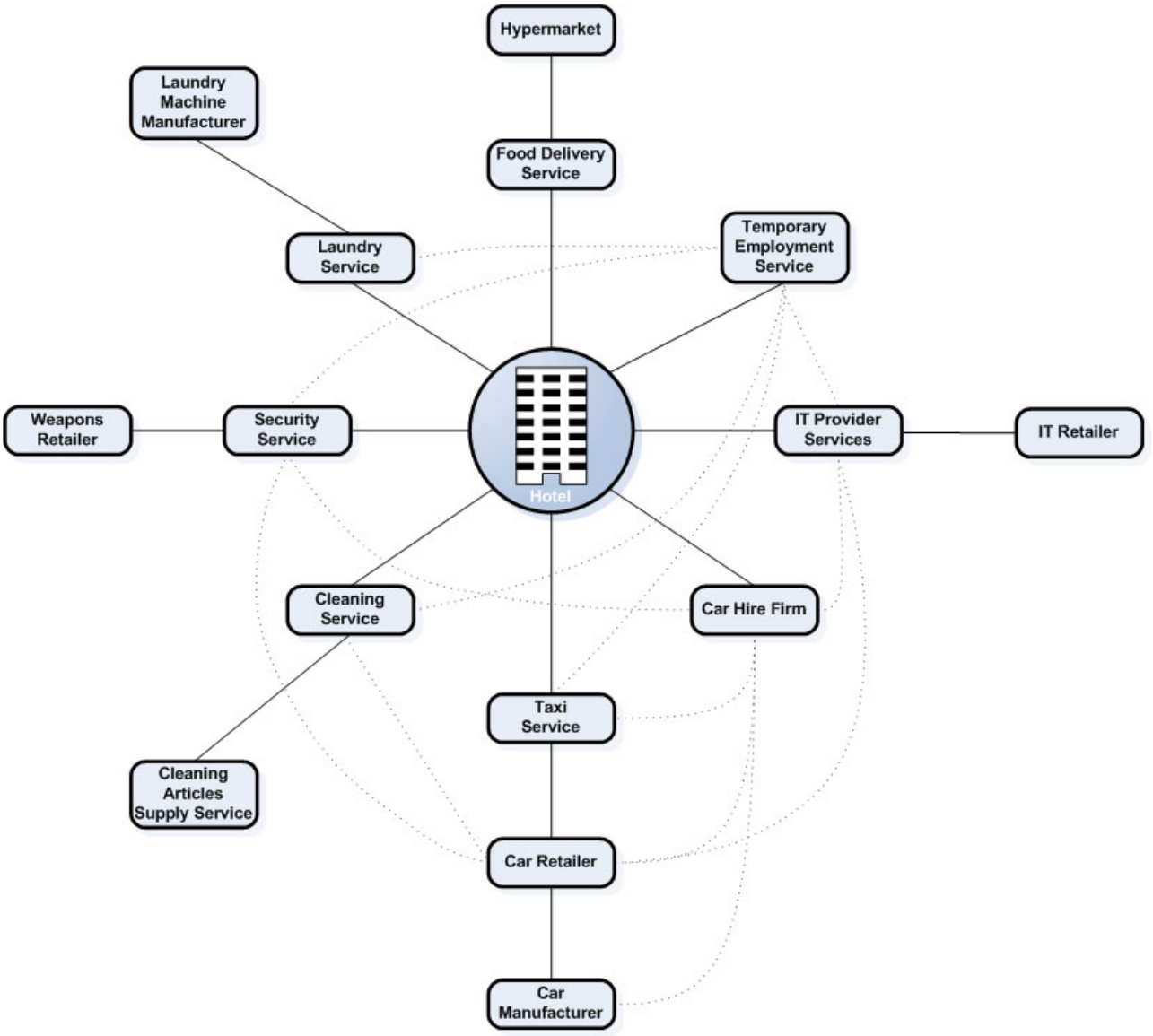


Figure 3: High-Level Example for a DBE Opportunity Space

The graphic is a very easy and uncompleted example of how an Opportunity Space could look like and what interconnections between originally 'hotel' related entities exist. On the first look the hotel offers a service like "car rental" provided through the entity "car hire firm" which is directly linked to the entity 'hotel'. Expanding this direct link to a more loosely level reveals further knowledge about connections

3.3 TECHNICAL INFRASTRUCTURE OF DBE USERS

As the term *Digital* Business Ecosystem implies, the concrete use of DBE components is only possible accessing the System on the digital path. This more or less important

constraint makes it necessary to define what requirements are needed to join the DBE and later on what is needed to become an active user in providing and requesting DBE services. Considering these aspects assures that accessibility to and usability from the DBE is guaranteed as far as possible. At first glance one could say that discussing these low level requirements is needless because any potential SME these days already matches the minimum requirements to join the DBE using an analogue modem integrated in a low-end personal computer with acceptable hard-drive space and running an externally hosted website. But especially taking more complex DBE services (probably coming up in the future) and applications into account a certain level of connectivity and availability to the web appears useful. Due to the existing DBE infrastructure and its involved components it is necessary to gather information from SMEs about what kind of IT-infrastructure they use to perform their daily business. For user profiling purposes the storage of SME IT-infrastructure information will be useful for securing compatibility of a requested DBE service and the IT an SME uses. Imagine a case where the functionality of a web service is requested but the service can only be executed using a certain operating system or under other specific conditions. Storing the basic IT-infrastructure within a user profile could avoid the problem of requesting services which are not compatible to a SMEs technical environment. Although these are still theoretical considerations it should be kept in mind when designing the user profile.

As a result of the interviews carried out by Task B4 so far following basic information regarding the IT-infrastructure of an average SME[1] could be collected. In spite of this it should be taken into account that this data cannot be guaranteed being generally representative facts. Nevertheless following results will give an impression on what informational and communicational level probably the majority of all potential DBE Users currently are.

Item	**Type**	**# SMEs**
Connection	*ADSL*	*7*
	Modem	2
	WiFi	3
	Radio	1
Technological Platform	*Windows NT / 2000*	*9*
Database	*MS Access*	8
	MS Excel	2
	mySQL	1
Security	Firewall	1
Office Software	*MS Office*	*9*
	Acrobat Reader	1
Internet	*Email*	*9*

Table 1: Basic IT Configuration of an SME

[1] The interviewed SMEs are all situated within the chosen Opportunity Space „Tourism“. Therefore there can’t be a prediction on what IT-Infrastructures SMEs in another Opportunity Spaces own.

As seen in Table 1 the majority of all interviewed SME uses ADSL for the connecting to the internet. The common Microsoft products like Win NT/2000, MS Office and Access can also be found in almost every interviewed SME. Interesting is, that only one SME has installed a firewall but shut it off due to performance reasons. As a result of this non of the interviewed SMEs uses any security applications like firewalls when connecting to the internet.

3.4 INFORMATION SOURCES

Information on users, service providers as well as service customers, are fundamental on designing connections and relationships between potential DBE participants. Gathering as many user data as possible is important for setting up prototyping profiles which provide the opportunity to test what information is really needed and what data has no further impact on the expressiveness of an user profile. The main sources for this approach were individually arranged interviews with representative SMEs within a 'Tourism' Opportunity Space. As this is already done by Task B9 - Use Case Definition – the collected information provides a basis for building aforementioned prototype user profile. The user profile itself probably won't need all information especially described by the BML Metamodel, designed by Task B18. The reason for this is a rapidly increasing complexity when too much data has to be taken into account for further profiling activities. A lot of information causes a lot of dependencies and therefore a too complex user profile could become quite difficult in using matching, as it maybe could become to complex measuring all dependencies.
When asking the question, *how* do we get user information and *where* do we get it from to initiate and maintain a long term business relationship a most frequent given answer is: from explicit and implicit user knowledge. Although the following techniques are described more detailed in Chapter 4.3 an overview should also be given at this place anyhow:

Explicit User Knowledge (Knowledge Based Information)

- *Questionnaires / Surveys*
  The user fills out general or topic related questionnaires consisting of explicitly asked questions and a given choice of answers.
- *Interviews*
  The user is interviewed by another person who collects information asking questions and aggregating the answers to profile usable information.

- *User Feedback*
  The user gives direct feedback on various concepts for example pressing a button “enough” which is interpreted as a negative feedback or a button “more” meaning a positive reaction.
- *Manual Profile Care*
  The user has the possibility at any time to add, delete or maintain personal information in his profile by accessing it manually.

Implicit User Knowledge (Behaviour based Information):

- *Ranking Techniques*
  Assumptions can implicitly be made based on the user’s behaviour in choosing items from a given ranked list in order to keep his decision in mind for further rankings.
- *Log File Analysis*
  The gathering and aggregation of information as records derived from user activity within the DBE environment.
- *Pattern Detection Mechanisms*
  Pattern Detection is similar to a market basket analysis but performed over time with a panoramic view of search and purchase patterns, especially considering web services in this context, to help to predict future customer behaviour.

Especially from the user profiling point of view a differentiation of the information which is to be stored appears necessary. Additionally one has to take into account that the accessibility level to the respective data differs from private via public to global. We differentiate between:

- *General User Information*
  Includes any data belonging to the user, its business data including business contact information and the available IT-infrastructure.

- *User Settings*
  The presentation level of the user profile can be adjusted to the user’s personal needs. Additionally settings like the level for accessing personal data can be defined. Using information rights management the user can decide what information stored in his profile will be available for the public and what will be kept private.

- *User Preferences*
  - Manually Set Preferences
    The User has the possibility to access his preference list at anytime to update and maintain his data by hand. In doing so, he can assign new metrics to attributes or add / delete data in a quite non-constraining way.

- Predefined Request List
  Within this List, the User has the possibility to predefine request statements in order to process request statements more efficiently. This technique can also be used to provide user preferred statements to other DBE components like the Recommender or the Evolutionary Environment.
- Manually Set Black List
  Additionally to his preference list the user can also dispose of a personal Black List which can contain information on what a user dislikes.
- Profiled Preferences
  This data is derived from user profiling methods and mechanisms used to collect information related to the user's behaviour in using the DBE.

- *User History* (optional)
  The objective of the user history is to collect and store user activities in a sorted list for further processing issues. This should be an optional feature in the first step because it is not clear yet how much data will be in a history of a user profile.

Additional user data examples are shown in Appendix 6-1.

3.5 DIMENSIONS AND METRICS

Measuring is the assignment of numbers or characters to real world entities. Therefore consistent rules are needed. Thinking of Quality of Service (QoS) requirements for software adequate and standardised measuring mechanisms are still missing. The concrete measurement of mechanic components (dimensions could be e.g. Roughness, Tolerances, Breaking Strength etc.) appears much easier than measuring Portability, Efficiency or System Stability of software components. Although those problems in measuring software components call for high carefulness, measuring methods are essential for increasing quality. A *metric* is "a quantitative measure of the degree to which a system, component, or process possesses a given attribute." [IEEE90, p. 47]

Measuring always requires the use of base items. Metering of these base items should be based on following *principles*:

- rules for metering are determined before beginning data collection
- metrics are arranged in an unambiguous way (consistent and objective)
- metrics should be based on the theoretical fundament of the matter
- metrics should be designed to be as easy as possible and computable

- empirically and intuitively convincing
- an effective mechanism for increasing quality management

For the *implementation* of metrics and the collection of measurable data some rules should also be take into account:

- automation of transaction is preferred
- use statistical analysis techniques for quality check-ups
- enhance comparability by using guidelines (e.g. coding rules)

Be it within a software component or in a recommendation process both need suitable metrics to measure quality. For illustration purposes further descriptions and examples are related to software components. Of course there are a lot of other possibilities defining metrics also for the recommender or any other DBE related component.

3.5.1 Metric Types

One can differentiate between two kinds of metrics:

- **Direct Metrics**
  These metrics (base terms) are usable without further interpretation. Examples could be: Costs, Lines of Code, Performance, Fault Rate etc.

  Example for direct metrics [Klüv01]:

	Project 1	Project 2	Project 3
Working Hours (man-months)	26	62	43
Employees	3	5	6
Costs (in 1.000 €)	330	880	630
Code (**K**ilo of **L**ines **O**f **C**ode)	12	27	20
Documentation (Pages)	365	1224	1050
Errors within the first year	29	86	64

  Using these base terms following further statements can also be formulated depending on the base term KLOC:

$$\text{Productivity} = \frac{KLOC}{man-months} \qquad \text{Quality} = \frac{errors}{KLOC}$$

$$Costs = \frac{Euro}{KLOC} \qquad \text{Ducumentation} = \frac{Pages}{KLOC}$$

- **Indirect Metrics**
  With this kind of metric a certain interpretation level influences the measurement especially when measuring parameters like Quality, Functionality, Efficiency, Reliability and so on. The function point method is very popular for this matter.

With this method the dimension e.g. Productivity could be measured considering predefined basic conditions within the system. Keeping up the direct metric example Productivity is now measured considering functionality provided by the function point method instead of the KLOC. In other words, the user influences the measurement using weights and ratings aligned to his preferences. Thinking of this method for user profiling purposes, it is also possible to express user preferences to a specific level by using this method. Further research on this will be done.

3.5.2 Preferences

Suitable dimensions and metrics for measuring user profiling attribute's interactions are fundamental for realising personalised rating, matching and recommendation activities. The importance of these variables arises from the fact that as many attributes as possible delivered to a user profile need to be comparable to process user requests adequately. If a customer, for example, is looking for a hotel near the exhibition centre in a foreign city, the location related adjective 'near' could be measured with the dimension 'distance'. The meaning of 'near' is described by a mapping interval like: 'near' → [0 – 0,5 km). As Figure 4 shows the request delivers three ranked hotels depending on their distance to the convention centre. The user can now choose the 'nearest' hotel to make his reservation.

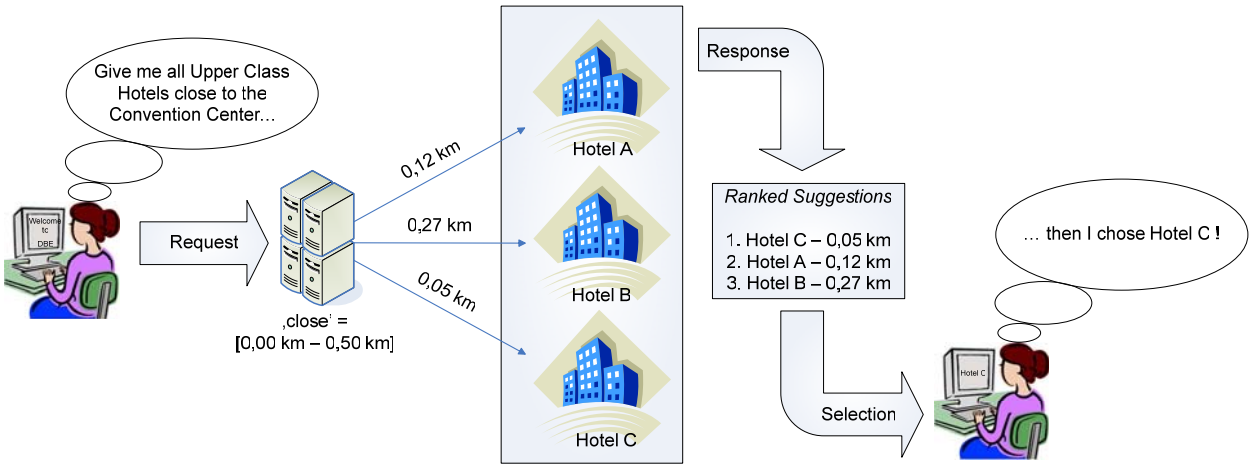


Figure 4: Practical use of a metric

Of course this is an easy B2C example and in this case the user could also enter a specific number in order to proceed the search but imagining a more complex service measuring variables are not easy to understand by an average user to be entered manually. The same approach can also be used when building B2B relationships. Let's assume a hotel tries to find a web service which provides credit card verification. Asking the person who is responsible for the IT of the hotel for the needed availability of this third-party service in a concrete mathematical way, than the answer would probably be '100 percent' because the

best performance is almost always wanted. A matching with this search criteria could fail, as for example only a web service with the availability of 99,8 percent exists which is still excellent. Another problem is that especially when talking about the availability of web services the scale runs extremely non-linear. This means that a value of 90 percent could probably mean to an average user that the availability of the web service is still very good but in fact it isn't. To avoid this situation it is easier to hide the mathematical meanings of metrics behind natural speech synonyms. In doing so a user just asks for an 'excellent' web-service availability not taking care of any mathematical statements and its specific meanings. This technique requires a mapping which will be the basis for a high usability at the end. The values shown in the following table are examples and are to be refined in testing phases. These variables and dimensions could be used for rating and matching activities:

Dimension	**Unit**	**Interval**	**Metric**
Distance	Kilometer / Miles	[0,0 - 0,5) [0,5 - 1,5) [1,5 - 5,0) [5,0 - 10) [10 - 30)	Walking Distance / Close Around / City Area Rural Suburban Far Away
Price	Currency in Euro € (in Million)	[0 - 45) [45 - 90) [100 - 300) [300 - ∞)	Cheap Affordable Expensive Luxury
Annual Turnover	Currency in Euro € (in Million)	[0 - 2) [2 - 10) [10 - 50)	Midget SME Small SME Large SME
SME Size	Employees	[1 - 10) [10 - 50) [50 - 250)	Midget SME Small SME Large SME
Hotel/Restaurant	Stars	[0 - 1] [2 - 3] [4 - 5] [6 - 7]	Lower Class Middle Class Luxury Class Decadent Class
Travel Agency	Employees	[1 - 5) [5 - 15) [15 - ∞)	Small Middle Big
Consumption per Person	Currency in Euro € (in Thousand)	[0 - 2) [2 - 10) [10 - ∞)	Little Middle Much
Air Ticket	Currency in Euro €	[0 - 250) [250 - 750) [750 - ∞)	Cheap Affordable Expensive
Loudness	Decibel - dB(A)	[15 - 20) [20 - 35) [35 - 50) [50 - 70)	Very Silent Silent / Whisper Comfortable Noisy / Busy

		[70 - 90) [90 - 120)	Loud Very Loud
Tour Distance	Kilometer / Miles (in Thousand)	[0 - 2) [2 - 10) [10 - ∞)	Small Middle Long
Tour Period	Days	[1 - 3) [3 - 7) [7- 14) [14 - ∞)	Weekend Short Middle Long
Total Revenue/Year	Currency in Euro € (in Million)	[0 - 5) [5 - 10) [10 - ∞)	Low Middle High
Growth	Change of revenue in Percent (%)	[20 - ∞) [0 - 20) (-100 - 0)	Fast Slow Decrease
Range of Service	Number of Countries	1 [2 - 15) [15 - 120)	National Regional Worldwide
Traveler State	Age (in Years)	[1 - 3) [3 - 13) [13 - 20) [20 - 30) [30 - 55) [55 - ∞)	Infant / Baby Child Teenager Young Adult Middle-Age Old
Traveling Group	Number of Persons	[3 - 10) [10 - 25) [25 - ∞)	Small Middle Big

Table 2: Examples for Metrics and their corresponding Units, Dimension etc.

3.5.3 Web Service - Quality of Services (QoS)

As seen in the B2B metric example in Chapter 3.5.2, Web Services will also play a role in the DBE. Therefore QoS requirements for web services, especially the quality aspects of Web Services will be highlighted in this section. QoS may include performance, reliability, scalability, capacity, robustness, exception handling, accuracy, integrity, accessibility, availability, interoperability, security, and network-related QoS requirements. These requirements are also quite important dimensions as they can also be stored within a user profile to represent what service levels are related to offered or booked web services. Because there are no real standardised metrics existing to the following dimensions adequate measures for the DBE purposes have to be defined [W3C03].

- *Performance*
  The performance of a web service represents how fast a service request can be completed. It can be measured in terms of throughput, response time, latency, execution time, transaction time, and so on. Throughput is the number of web service requests served in a given time interval. Response time is the time required to complete a web service request. Latency is the round-trip delay (RTD) between sending a request and receiving the response. Execution time is the time taken by a web service to process its sequence of activities. Finally, transaction time represents the time that passes while the web service is completing one complete transaction. This transaction time may depend on the definition of web service transaction. In general, high quality web services should provide higher throughput, faster response time, lower latency, lower execution time, and faster transaction time

- *Reliability*
  Web services should be provided with high reliability. Reliability here represents the ability of a web service to perform its required functions under stated conditions for a specified time interval. The reliability is the overall measure of a web service to maintain its service quality. The overall measure of a web service is related to the number of failures per day, week, month, or year. Reliability is also related to the assured and ordered delivery for messages being transmitted and received by service requestors and service providers.

- *Scalability*
  Web services should be provided with high scalability. Scalability represents the capability of increasing the computing capacity of service provider's computer system and system's ability to process more users' requests, operations or transactions in a given time interval. It is also related to performance. Web services should be scalable in terms of the number operations or transactions supported.

- *Capacity*
  Web services should be provided with the required capacity. Capacity is the limit of the number of simultaneous requests which should be provided with guaranteed performance. Web services should support the required number of simultaneous connections.

- *Robustness*
  Web services should be provided with high robustness. Robustness here represents the degree to which a web service can function correctly even in the

presence of invalid, incomplete or conflicting inputs. Web services should still work even if incomplete parameters are provided to the service request invocation.

- *Exception Handling*
  Web services should be provided with the functionality of exception handling. Since it is not possible for the service designer to specify all the possible outcomes and alternatives (especially with various special cases and unanticipated possibilities), exceptions should be handled properly. Exception handling is related to how the service handles these exceptions.

- *Accuracy*
  Web services should be provided with high accuracy. Accuracy here is defined as the error rate generated by the web service. The number of errors that the service generates over a time interval should be minimized.

- *Integrity*
  Integrity for web services should be provided so that a system or component can prevent unauthorized access to, or modification of, computer programs or data. There can be two types of integrity: data integrity and transactional integrity. Data integrity defines whether the transferred data is modified in transit. Transactional integrity refers to a procedure or set of procedures, which is guaranteed to preserve database integrity in a transaction.

- *Accessibility*
  Web services should be provided with high accessibility. Accessibility here represents whether the web service is capable of serving the client's requests. High accessibility can be achieved, e.g., by building highly scalable systems.

- *Availability*
  The web service should be ready (i.e., available) for immediate consumption. This availability is the probability that the system is up and related to reliability. Time-to-Repair (TTR) is associated with availability. TTR represents the time it takes to repair the web service. The service should be available immediately when it is invoked.

- *Interoperability*
  Web services should be interoperable between the different development environments used to implement services so that developers using those services do not have to think about which programming language or operating system the services are hosted on.

- *Security*
  Web services should be provided with the required security. With the increase in the use of web services which are delivered over the public Internet, there is a growing concern about security. The web service provider may apply different approaches and levels of providing security policy depending on the service requestor.
  Security for web services means providing authentication, authorization, confidentiality, traceability/auditability, data encryption, and non-repudiation. Each of these aspects is described below.
  - Authentication
    Users (or other services) who can access service and data should be authenticated.
  - Authorization
    Users (or other services) should be authorized so that they only can access the protected services.
  - Confidentiality
    Data should be treated properly so that only authorized users (or other services) can access or modify the data.
  - Accountability
    The supplier can be hold accountable for their services.
  - Traceability and Auditability
    It should be possible to trace the history of a service when a request was serviced.
  - Data encryption
    Data should be encrypted.
  - Non-Repudiation
    A user cannot deny requesting a service or data after the fact. The service provider needs to ensure these security requirements.

3.5.4 SME Categorisation – The Rathbone Code

The user profile will also include a categorisation of the respective SME using the categorisation code designed by Neil Rathbone working on DBE Work Package 15. The reasons and backgrounds why the code has been developed can be gleaned in WP15 and therefore is not supposed to be the subject of this deliverable. The following passage only deals with the concrete categorisation code and an example on how this will be used within the Profile. The Rathbone Code bases on three fundamental factors when categorising an SME [Rath04a, Rath04b]:

- Value

  The way that companies add value determines what is their most important ‘asset’ and it is around this that they organise their culture, management and systems.”

  (“How does the company create or add value?”)

- Growth Stage

  Assuming that their business is successful, companies tend to grow distinct stages. It is possible that so called ‘pain barriers’ (e.g. Long hours worked by all staff, Systems outgrown/overloaded, Key turnover bands reached etc.) can be the consequences if a growing company runs our of management resources.

  (“Which stage of life is the company at?”)

- Personality

  Most SMEs are not neutral profit-generating machines, but are driven by the personal ambitions and views of its owners and managers. The ownership and management of a company to some extend imposes its own ‘world view’ onto an SME.
  (“Which type of ownership and management has it got?”)

A specific categorisation code of an SME can be stored within the profile. Following scenario example shows how the Rathbone Code works:

“An SME is a family-owned enterprise, has 30 employees and already exists in the 3rd generation. Within the last 20 years the annual turnover continuously increased every year. The future prospects are very well. The SME manufactures pencils using modern computer technology. Nevertheless the staff has to work a lot and does long hours.”

Using the categorisation process and its selection criteria defined in [Rath04b], the Rathbone Code (see also Appendix 6-2) delivers following individual code for the example SME:

Selection Criteria	**Code**
The SME manufactures pencils using Batch Processes	C2(M)
The SME has one manager who isn’t a worker. A Pain Barrier could be the long working hours of the employees.	2+
The SME is a high growth family business as the turnover increases 20 percent in average per year.	H3

Table 3: Rathbone Code Example.

The code “C2(M) – 2+ - H3” can now be stored in the user profile. The categorisation is an additional information source next to the BML description of the SME. The user profile only cares about the storage of a given code and is not responsible for the categorisation process as such.

4 USER PROFILING APPROACH

4.1 GENERAL FIELDS OF WORK

The user profiling work package as such includes three main aspects which have to be taken into account when designing the basic profiling structure.

1) What kind of data is needed for a profile

   This aspect deals with the question what kind and what types of data are suitable and meaningful to be included into a user profile. Obviously not every information of and from a user needs to be stored within a profile as this would probably result in an information overhead. So the question will be what information is really needed. It goes without saying that this deliverable can only give a jump start on what specific information on preferences and attributes could be saved in a profile and therefore cannot be exhaustive. An overview on possible preferences and metrics is given in Chapter 3.4 and 3.5.

2) What mechanisms are needed to collect data from 1)

   To provide the profile with aforementioned information adequate mechanisms will be necessary. These will be represented in the context of the upcoming papers. A basic introduction to possible knowledge-based and behaviour-based information retrieval possibilities is given in Chapter 4.3.

3) How to organize and represent this data within a user profile in an intelligent way

   The organisation of the collected and selected profile information is a not to be sneezed at aspect. Possibilities have to be found for sorting, ranking and representing a user profile’s set of data in a way which finally offers computable information to other DBE components like the Recommender or the Evolutionary Environment. A first idea on how this dynamic representation could be realised is

shown in Chapter 4.4. An easy example in the Appendix 6-3 gives an impression on how the structure of information within a profile can basically be represented.

4.2 User Profiling – Definition and Differentiation

As the expression user profiling already implies, the main objective of this mechanism is to create an individual user related knowledge base adapted from the user's explicit and implicit activities within the Digital Business Ecosystem. Taking knowledge-based and behaviour-based activities into account it should be possible that user characteristics and preferences mapped in an individual user profile will exemplarily support the recommendation process of useful personalised services in evidence. Personalisation is a means of meeting the customer's needs more effectively and efficiently, making interactions faster and easier and, consequently, in creating customer satisfaction.
Since there were some obscurities regarding the role and tasks of user profiling, especially compared to other DBE components like the Recommender, following differentiation should be done for clarification. A main task of the DBE user profiling will be the mapping of the real world user's character and its preferences to a virtual user profile. This means that information about a user should be stored within a profile as suitable and as fine grained as possible.

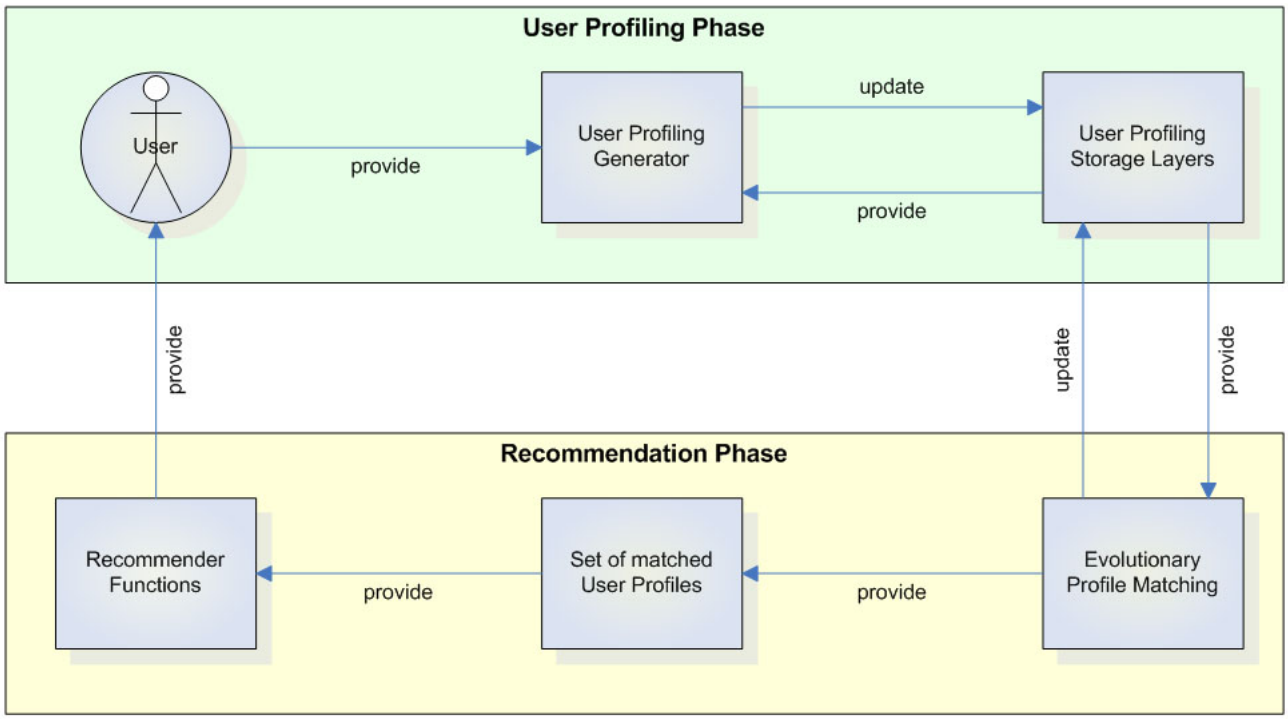


Figure 5: User Profiling Life Cycle Process

If we assume that the collected user profile information nearly completely corresponds to the intentions and characteristics of the real user, the personalisation process could be improved significantly. The user profiling life cycle again clarifies the role of the user profiling in comparison to the Recommender. The main function is to organise, represent and store user relevant information. The so called User Profiling Generator will be responsible for implementing the mechanisms for collecting user information and the User Profiling Storage Layers will dynamically organise user relevant preferences derived manually by the user or automatically through the Profiling Generator. A more detailed view will be given in Section 4.4.

4.3 USER PROFILING OPPORTUNITIES

The term user profiling opportunities refers to the kinds of information delivered to a user profile. This also implies possible mechanism on a higher level to gain user information.

4.3.1 Knowledge-Based Representation (static)

The profile is fed with data coming from explicit input directly from the user. Examples for this method of raising information are:

- *Interviews and Surveys*
  They describe a series of orally-delivered questions designed to elicit responses concerning attitudes, information, interests, knowledge, and opinions. Interviews may be conducted in person by telephone or using Surveys. The three major types of interviews are [EvCe04]:
    - *Structured*, where all questions to be asked by the interviewer are specified in advance
    - *Semi-Structured*, where the interviewer can ask other questions and prompts in addition to the specified questions
    - *Unstructured*, where the interviewer has a list of topics, but no or few specified questions

  Explicit Information about the user in the DBE will mainly be derived by using web-based forms which include a set of structured questions. Usually the user can choose an option from a list of options to submit his preference and the desired

level of importance. Another way to get basic information about the User is making use of an interviewer who has a much higher technical and operational background of the functions and possibilities of the DBE. This role could for example be adopted by a regional catalyst which holds an unstructured or semi-structured interview, summarises the given answers and finally delivers a customised set of data to the interviewees' user profile. This will be the case if a DBE joining SME won't have the time or skills for providing this information on their own by using forms. This activity will usually be used in the initial creation of the profile.

- *Data Updates*
  This technique is completely based on the user's activity on updating and maintaining his profile. He manually navigates through the profile and adds or even deletes profile data. The activity on this technique depends on the user's intentions in refining his personal preferences and information.

- *Explicit User Ratings*
  The rating could be interpreted as an extension to the aforementioned Data Updates. The User redefines the ranking of preferences manually in a way he likes. Ratings which were added automatically using profiling mechanisms can be overwritten if the user does not agree with it.

4.3.2 Behaviour-Based Representation (dynamic)

Additionally to the knowledge-based information representation the dynamic interaction between the user and the DBE can be taken into account using behaviour-based approaches. There are many possibilities how information can be gathered from a person navigating through the DBE, providing and requesting services. Common approaches among others are:

- *Log-File Analysis*
  The data available in web environments is three-fold and includes server data in the form log files, web metadata representing the structure of a web site, and marketing transaction information, which depends on the products and services provided. All server software applications, such as email, www and ftp kept records of requests and actions taken in a log file. Following a short definition should give an impression of what kind of format log files could be. "Given a set R of report elements and a distinguished subset $K \subset R$ of keywords, we define a report as a finite sequence of

report elements beginning with a keyword or with an ordered pair [timestampelement; keyword].” [Andr98a, Andr98b]

Regarding to [Vald01, p.7] log files and their usage can be described as follows:

- The log file is an auxiliary output file, distinct from other outputs of the program. Almost all log files are plain text files.
- Starting up an application, the log file is either empty, or contains whatever was left from previous runs of the application.
- During program or application operation, lines (or groups of lines) are gradually appended to the log file, never deleting or changing any previously stored information.
- Each record (i.e. a line or a group of lines) in a log file is caused by a given event in the program, like user interaction, function call, input or output procedure etc.
- Records in log files are often parameterised, for example they show current values of variables, return values of function calls or any other state information.
- The information reported in log files is the information that programmers consider important or useful for program monitoring and/or locating faults.

The analysis of these log files is a broad level view of who is using various technologies, how frequently and for what purposes. Using this technique it is possible to collect implicit information on the user’s behaviour storable in a user profile. Of course privacy issues play a big role in this context, but are not the subject of this basic mechanism.

- *Implicit User Ratings*

  The general idea of this rating technique is to dynamically adjust user preferences depending to their behaviour in using the DBE. This means that preferences which were originally rated by the user by hand can automatically be changed by the system when selected criteria occur. Additionally to the automatic modification of user preferences the implicit user rating is supposed to add new preferences and attributes if predefined values and thresholds for adding new attributes are reached. The concrete implementation of this approach can be done with the help of profiling mechanisms consisting of specific profiling algorithms being utilised in the further course of the project.

4.3.3 Possible conclusions drawn from a DBE User Profile

An interesting aspect on the usefulness of a user profile is, what kind of conclusions can be drawn from the existing profile data. Depending on the stage of maturity of the whole

DBE project and the increasing number of participating users and correspondingly user profiles following different phases of information interpretation derived from a set of profile data can be introduced:

- Phase 1 (atomic user profile)
  - general history-driven user interests (Surfing History, Purchase History etc.)
  - Preferences and assumed dislikes

  In this phase, conclusions on the user's interests can only be done by analysing information explicitly written in an atomic user profile. This means that it is not possible in this phase to enrich a user profile with additional information besides some behaviour-based knowledge using specific mechanism.

- Phase 2 (clustered user profiles)
  - self-adapting user profile depending on other similar user profiles (opportunity space cluster, location cluster etc.)
  - simple comparisons in user behaviour

  Here the basic idea is to cluster similar profiles using predefined clustering parameters. The objective is to compare profiles which are basically congruent in predefined preferences to finally offer the opportunity to make suggestions to a user like: *"There are several profiles in the opportunity space 'tourism' which are very similar to your current profile preferences regarding 'business flights'. These profiles also include following additional preferences: entertainment_system = preferred, time_to_connecting_flight < 2 hours, airport_transfer_service = yes. Would you also like to add these preferences to your profile to refine your personal preferences on 'business flights'?"* With this approach it should be possible to benchmark profiles and finally to derive statistical information from a clustered group of users for example in a certain region.

- Phase 3 (networked user profiles)
  - overall business models of SMEs
  - competitiveness of current SME Business Model

  A vision will be that at a certain stage of the project where enough users and user profiles exist and the information within a user profile or even within a clustered group of user profiles is as detailed as possible. Then it could be possible to map and match profiles on strategic areas like competitiveness of business models or similar. The validation of already existing business models to other SME profiles could also be an interesting approach. For example: *"There are six SMEs in your opportunity space which have very similar preference settings on their customers and they use the same software applications (ARIS-2000, CRM-Wizard…) in their daily business. Would you like to know more on this software?"* Even the creation of new business models could be an option using mechanisms coming from the

Evolutionary Environment. Of course anonymity, privacy and security are not taken into account in this vision yet but could be in an advanced phase of the project. The first step to this vision is the bespoken critical mass of user profiles and suitable information stored within them. How large this critical mass has to be is not predictable without empirical test which will maybe be possible in the end of the project phase.

4.4 USER PROFILE ARCHITECTURE

The main objective of the following user profile architecture is to represent profile information as dynamically as possible using suitable internal representation mechanisms cooperating with selected profiling mechanisms. A layer architecture was chosen to enhance the data representation within a profile. The main reason for this decision was that it seems reasonable to classify and sort data against their refresh period. With this technique aforementioned objective should be feasible. The central approach is an internal mechanism to label every data set with a time stamp giving information on when the data set has been created, when it was used the last time and when it will expire. The main reason to split the layers into short-term and long-term memory was to try to imitate the natural learning process of human beings in a very simple way. If we have a close look on ourselves and how we keep information in mind, than we also recognise that it makes a difference if we've seen or learned something once or if we repeated it a lot of times. As a consequence information can easily be remembered for a few days.

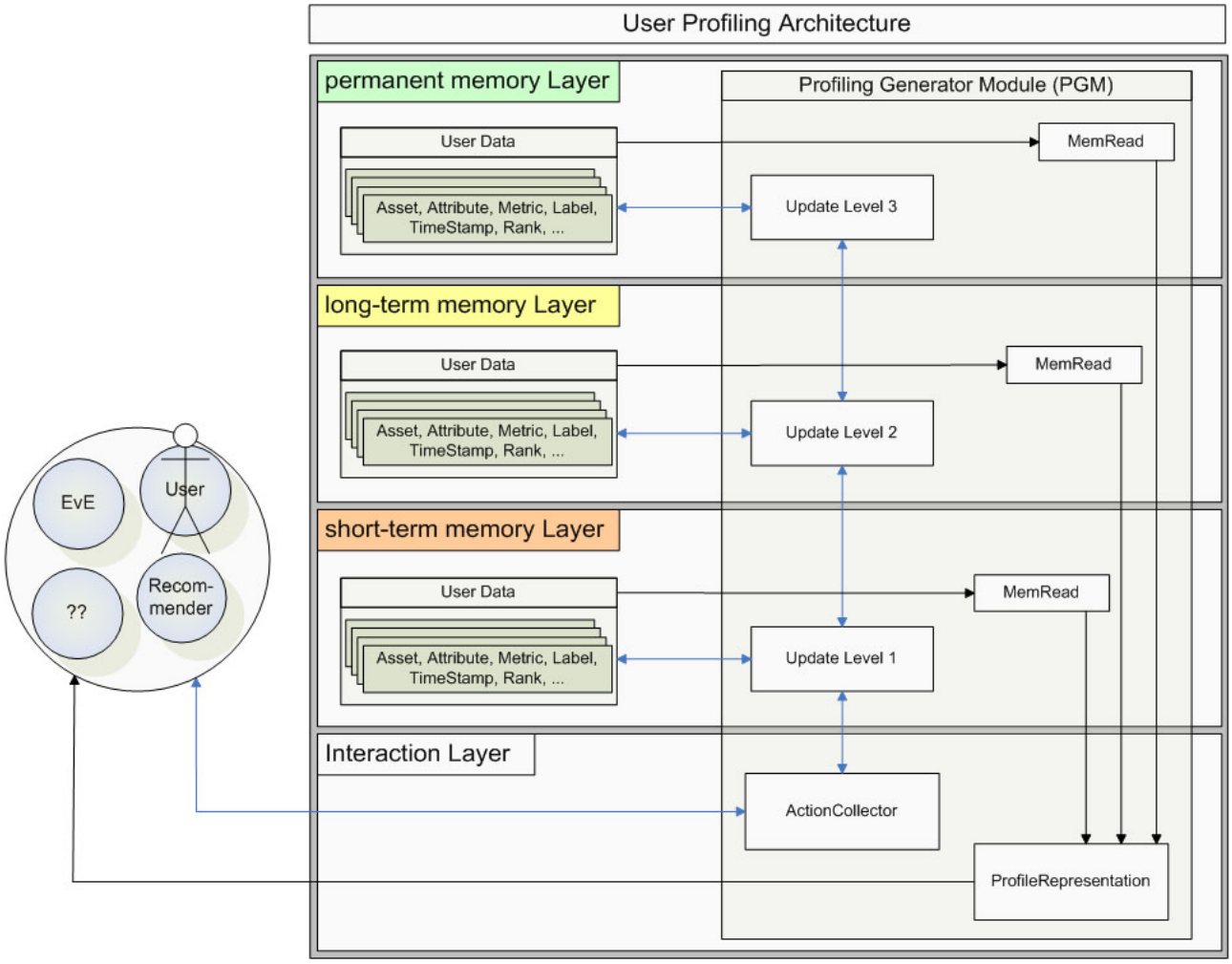


Figure 6: Basic User Profiling Architecture

But will this information still be available in our mind in lets say half a year without having repeated it in the meantime? And what about preferences which we have now but maybe won't have a few moths later? Of course this is a very easy example and not all facets of learning techniques for humans are taken into account, but it is assumed that this architecture could represent a user and his profile closer to his real world intentions. The concrete mechanisms will follow in a later part of the DBE project. Following parts describe the idea of a profiling layer and afterwards the briefly indicated labelling mechanism will be introduced.

4.4.1 Interaction I/O Layer

This layer is responsible for collecting profile information using profiling mechanisms. It is suggested to provide any input and output activities via the interaction layer to have a single point of contact to the profile and to reduce complexity in designing a user profile interface. The input data streams can either automatically be derived from any other components of the DBE like the Recommender, the EvE or Accounting or manually be

entered by the user after logging into his profile. This task is supposed to be done by an Action Collector or something similar. The internal representation of all user relevant data will be a simple XML file containing all relevant information. This is also the representation of the output information for other DBE partners. Another way could be a very easy representation of user relevant information within a user cookie. It still has to be clarified if a hybrid use of existing representation possibilities is useful or if an exclusive use will provide more benefit to the user profiling. This representation task could be realised by a Consolidator to provide profile information from all layers to an interface. As there probably will be a lot of information about a user stored in his profile in the future one have to think about if it makes any sense to provide a complete XML-output file to other components. A future step could be a generation of XML-output files depending on the specific request of the enquiring DBE component. In doing so processed information could be provided. This information can specifically be assigned to the needs of components like the Recommender or the EvE. This may become important in the future under performance aspects.

4.4.2 Short-Term Memory Layer

The layer generally contains data which was added recently to the user profile. This is similar to a filtering function to make sure that newly added information (especially automatically added data) doesn't get so much influence within the profile right from the beginning. Several mechanisms will be responsible to organise, rate and rank incoming and already existing information within the profile depending on several factors like e.g. the user's explicit ranking or his usage behaviour in the DBE. Assuming that each data set is labelled correctly with a timestamp, usage rate and so on, every incoming behaviour-based or self-adopted profile information will be stored directly in this temporary short-term memory including the usage rate and all other relevant data set information. The data set is assigned to a standard ranking factor which is used to weight an asset or a group of assets to represent the individual importance of an asset within this layer. If a given threshold is reached (e.g. a number of same used preferences existing in the short-term profile within a given time slice) this preference is moved to the long-term memory as it is assumed that this preference is very interesting to the user and therefore responds to his real preferences.

4.4.3 Long-Term Memory Layer

This layer is similar to the Short-Term Memory Layer besides the fact that preferences in this stage have a much higher weight within the whole profile. They have a higher rating-value and influence recommendations etc. more than preferences in the short-term memory do. Not only the weight and influence of a preference within a profile plays an

important role but also the use of this preference. At this point the timestamps become significant. The dynamic of the data organisation within the profile layers mainly depends on them. A preference which for example was automatically added to the long-term memory layer one year ago hasn't been used (in any way) until now. One could therefore assume that this preference is not interesting anymore to the user and maybe this assumption is correct. The internal mechanisms realise that and downgrade this preference to the short-term memory layer which lowers the value of the preference within the profile. The new expire timestamp has a much lower durability. In the case that even in this new time period the preference will not be used it will be set to the status invalid. So there won't be any influence by this preference to the profile at all. Of course the user has the final decision if he wants to delete the preference or to reactivate it.

4.4.4 Permanent Memory Layer

Within this layer information will be stored that, as the name already says, is ment to be of permanent nature. The term permanent refers to the idea that a data set should at best only be created once and if possible never be deleted or modified during the life span of a user profile. Of course this is a very hard constraint but as said before "if possible". Conceivable information about a user which could maybe be stored permanently or at least a very long time could be his username, date of birth, company name, country, and so on. This data could come from sources like the BML Editor where the user can specify almost every business related information. Additionally, manual inputs can be done by for example when initiating the user profile after registering at the DBE.

4.4.5 Profile Generator Module (PGM)

This module will contain the user profiling functionality, including three basic core mechanisms for

- organising preferences dynamically,
- making profile information available to other components and
- collecting user information depending on his usage of the DBE.

The functionality of the PGM stretches across all aforementioned memory layers. An algorithm will manage the organisation of the incoming and already existing user profile information using timestamps and markers. This technique can be used for managing at least an easy change process of user preferences. Of course profile information has to be made available to other DBE components in order to make them work more properly. It is not completely clarified, if other components will have the possibility to access user information at its basis, having control on atomic data sets or if access to personal user

information will only be possible by connecting to an interface, provided by the profile. This interface could make aggregated user information available to other DBE components. Especially when thinking about privacy issues this approach could be useful for the trust of the user in the security of the DBE because he can keep control on what data can be access externally and what data will stay private. The third core mechanism of the PGM will include selected profiling mechanisms for collecting information about the user and his behaviour within the DBE. The detailed description of these activities will be part of the next deliverables.

4.5 EXTENDED BML

The well known BML meta-model developed by ISUFI (WP 15) is used to describe any data on business information regarding a SME. Although the BML itself is very powerful and can be used for defining and describing respectively a maximum possible business information space it is not possible to define information outside the business field. Of course technical descriptions for example for Web Services are specified by other languages like SSL or SDL but nevertheless all languages are quite restricted to their own model. For user profiling purposes and especially the storage of all kinds of information the adoption of one language would have been too restrictive but the adoption of all languages would have probably become too complex to handle. Therefore the compromise was to extend the already existing powerful BML metamodel with user profiling specific functions to consider also technical data in this model. With this approach it should be possible to represent business related user information derived from the BML as well as IT and software related information derived from other technical languages like SDL or SSL. The following models are not meant to become an own meta-language (in fact they aren‘t ‘pure’ meta-models) but rather to combine information already existing within the DBE for user profiling purposes, especially when considering the representation of user profiling data.

4.5.1 Profile Package Dependencies

As following packages should be used as an extension of the already existing BML model a point of connection to BML is necessary. An approach could be the import of the *BusinessCore* in the BML metamodel to have access to all business related data. The *ProfileCore* imports the *BusinessCore*. The package *ProfileRequests* uses the packages *ProfileLabeling* and ProfilePreferences for expressing request statements. The package *ProfileLabeling* uses information of the package *ProfilePreferences* which is connected to

the basic package *ProfileCore*. With these packages it should be possible to express all user profile necessary data and to add functionality to each set of data.

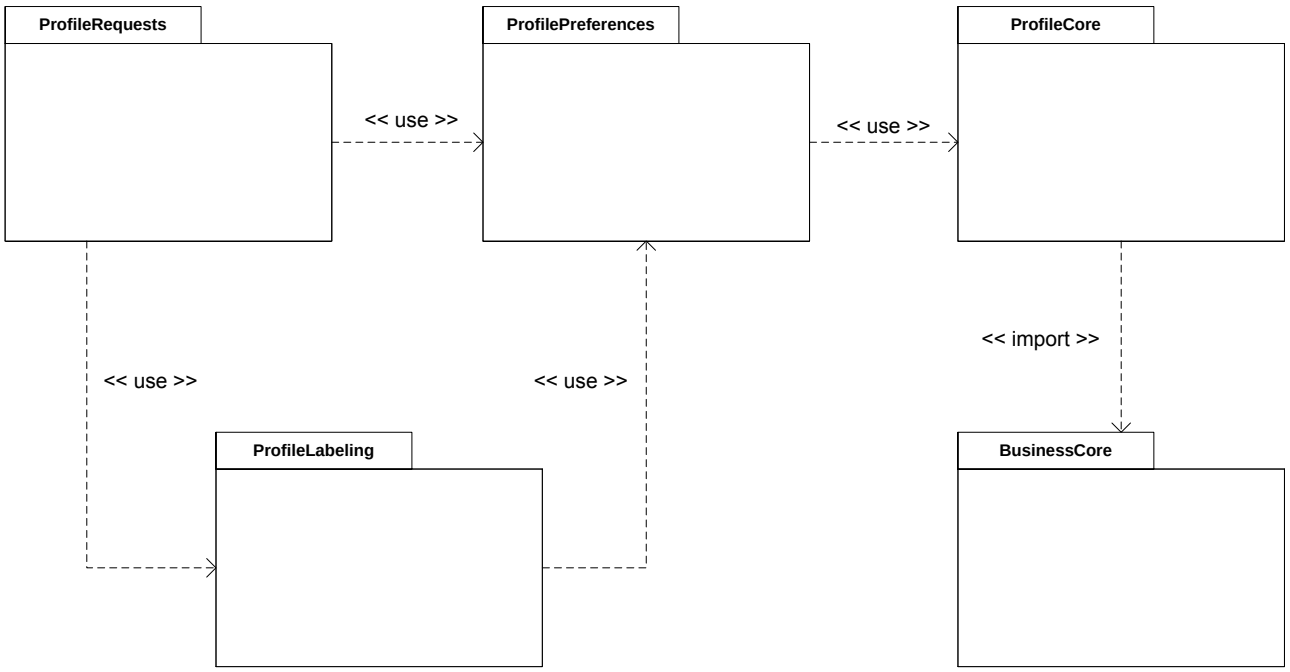


Figure 7: Profile Package Dependencies

The following sections provide a refined view on the user profiling packages.

4.5.2 Profile Core Package

The package *ProfileCore* is very similar to the package *BusinessCore*. The only difference is, that elements are not restricted to the business domain like in BML. Therefore a renaming from *BusinessElement* to *ProfileElement* seemed to be useful to express the unrestricted information representation approach of the user profile.

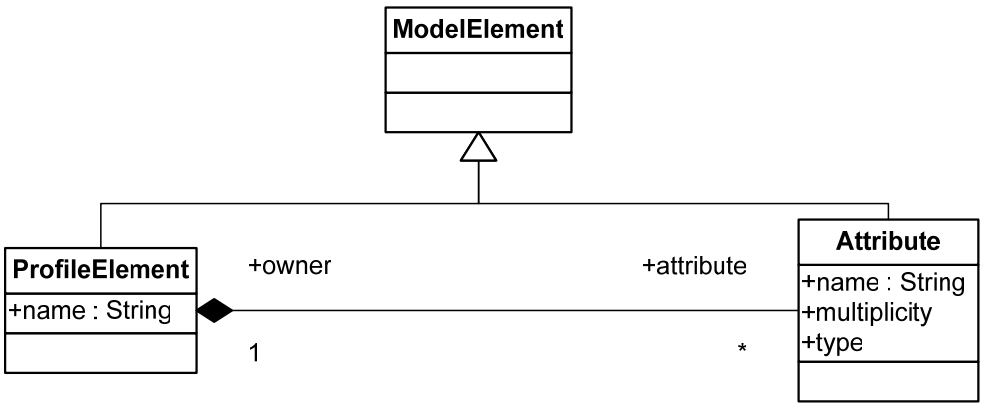


Figure 8: Profile Core Package

A *ProfileElement* can now be everything, not taking care of what domain the information is coming from (business, technical and so on. It has a name and is the owner of an attribute also having a name.

4.5.3 Profile Preferences Package

The package *ProfilePreferences* outlines the main relationship between the defined entity its assets and the involved preferences. The class *ProfileEntity* is a subclass of the *ProfileElement* which is basically defined in the *ProfileCore*. A *ProfileEntity* can own several subunits but doesn't have to. Usually an entity owns one or more assets to refine the entity in an arbitrary way.

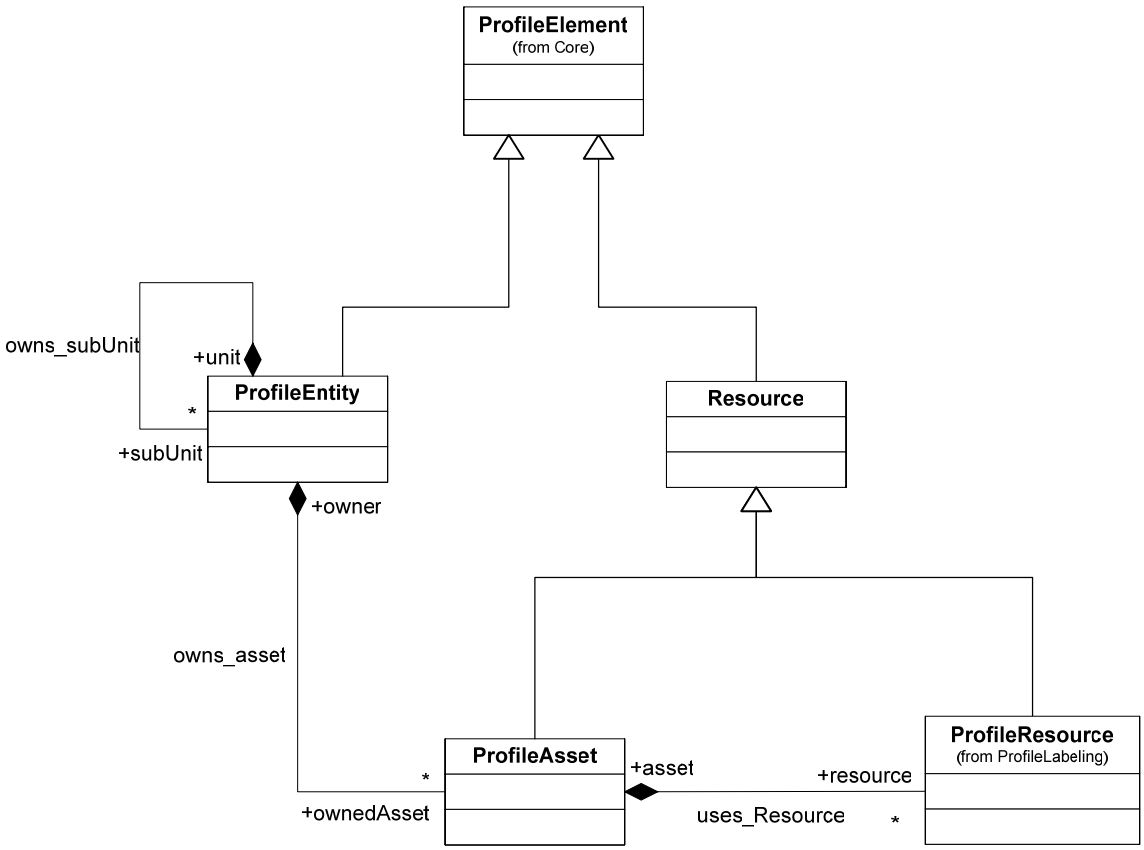


Figure 9: Profile Preferences Package

These assets are additionally described by using the class *ProfileResources* which can be found in the package *ProfileLabeling*. Both, the class *ProfileResource* and the class *ProfileAsset* are subclasses of the abstract class Resources. Following example will point up the function of the profile preference package classes.

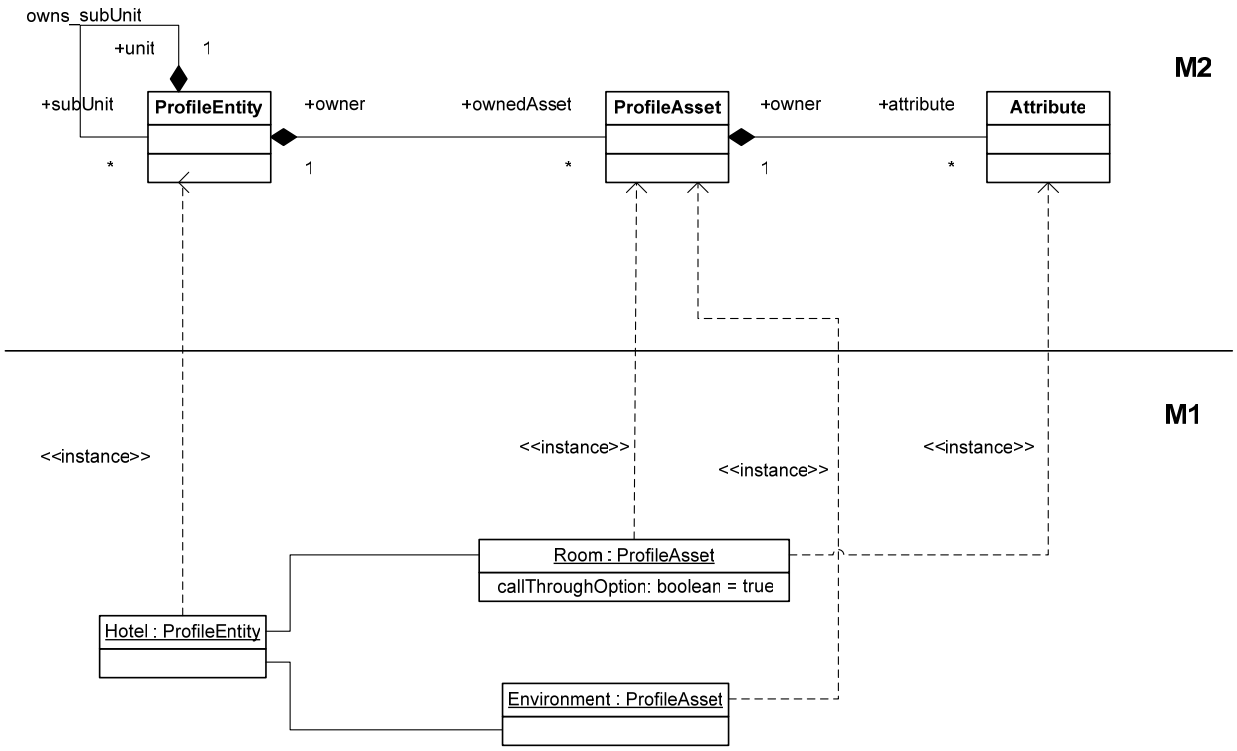


Figure 10: Profile Preferences Package - Example

An instance of *ProfileEntity* on the M1 Level could be a Hotel which owns the *ProfileAsset* room for further descriptions of preferred room specifications and the *ProfileAsset* environment to refine the geographical information space. Additionally a hotel room has an attribute *callThroughOption* which allows direct inbound and outbound telephone calls without using any hotel operator.

4.5.4 Profile Label Package

This package defines the core needs for labelling an asset's attribute with all necessary information to process and handle user profiling data in a suitable way. One could also say the 'functionality' of an attribute is provided by this package. Generally an attribute can be extended with information coming from the *ProfileResource* subclasses *Metric*, *Label* and *Ranking*.

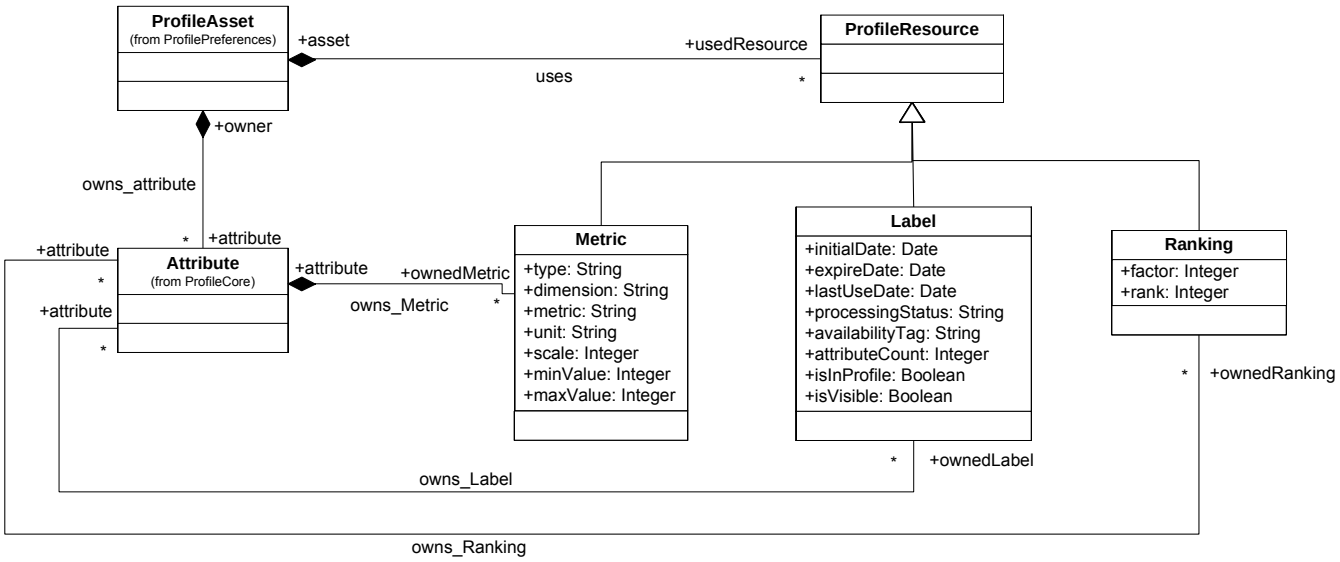


Figure 11: Profile Label Package

The class ***Metric*** adds the function of a metric to an attribute using following variables:

Variable	Function	Example (boolean)	Example (interval)
Type	Gives information of what type the metric is. The Boolean type is used if an attribute of an asset should be just valid or not (hard constraint). If weaker constraints are needed the user can use scales.	boolean (An asset could be a hotel room with the attribute non-smoking)	intervall (An asset could be the location of a hotel. The hotel shall be close to the city center)
Dimension	Describes the context the metric is used.	Availability	Distance
Metric	Assigns an expression or value to a specific scale.	'yes'	'close'
Unit	The unit defines what value the measurement is made of.	boolean	Kilometers
Scale	A number which represents the mathematical basis for the assigned metric.	0 = false 1 = true	1 = [0,0 – 0,5]
minValue	The lower bound of an interval	-	0,0
maxValue	The upper bound of an interval	-	0,5

Table 4: Profile Label – Class Metric

The class ***Label*** adds information regarding the usage and time to the attribute using following variables:

Variable	Function	Example
initialDate	The Date when the data set was started	01.02.2005
expireDate	The Date when the data set will expire	31.03.2005
lastUseDate	The Date when the data set was used the last time	02.03.2005
processingStatus	The status shows, if a data set was added or modified automatically by the system or manually by the user	manual
availabilityTag	This tag generally shows, if a data set is still valid or is already expired	valid
attributeCount	Shows how often a data set was accessed	2
isInProfile	Shows, if a data set exists in a profile	true
isVisible	Shows, if a data set is visible to a user	true

Table 5: Profile Label – Class Label

The class ***Ranking*** adds information regarding the ranking status of an attribute using following variables:

Variable	Function	Example
factor	The factor gives additional possibilities to rank data sets (especially when using profiling mechanisms)	2,6
rank	The position of a data set within a ranked list	3

Table 6: Profile Label – Class Ranking

The following example shows how a data set could be labelled using the Profile Label Package

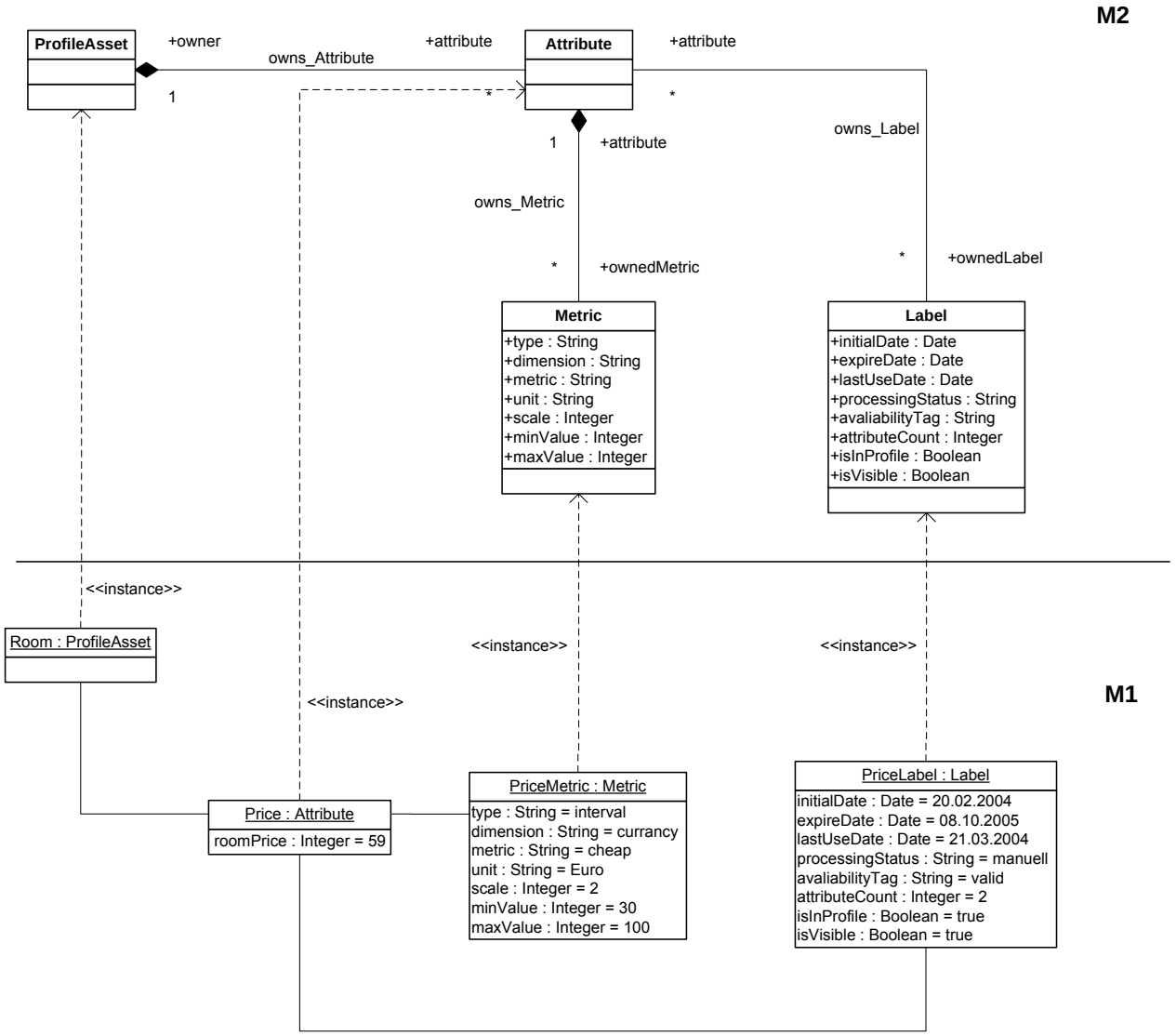


Figure 12: Profile Label Package - Example

The example shows how a metric and a label gets assigned to an asset. A Room usually has a price which is in this case 59 € per night. The associated metric is ‘cheap’ as it is assumed that a room price e.g. in a four star hotel is cheap when it moves between 30 and 100 Euros per night. The scale value 2 codes the interval [30, 100] to the metric ‘cheap’ to be able to realise a connection between these dependencies. Additionally this data set gets a label with the date of its creation, the so called *initialDate*. Depending on the user profile self-organising mechanisms an *expireDate* will manually or automatically be set. As the date of the current data set is not expired yet, the *availabilityTag* is set to be true. The processingStatus visualises if a data set has been created or modified manually by the user or automatically by the system.

4.5.5 Profile Request Package

The package can be used for representing and for expressing request statements. This basic approach (see following figure) will be a simple record of a request statement. An advanced approach in the further course of the project could be an integration of the QML (Query Modelling Language) for representing request statements. Further details on QML can be found in Deliverable D17.1. Additionally requests defined in SBVR (Semantics of Business Vocabulary and Business Rules) can also be stored within a profile as especially this kind of language will be used by the EvE as basis for further functionality aspects (see D9.1 and D15.1)

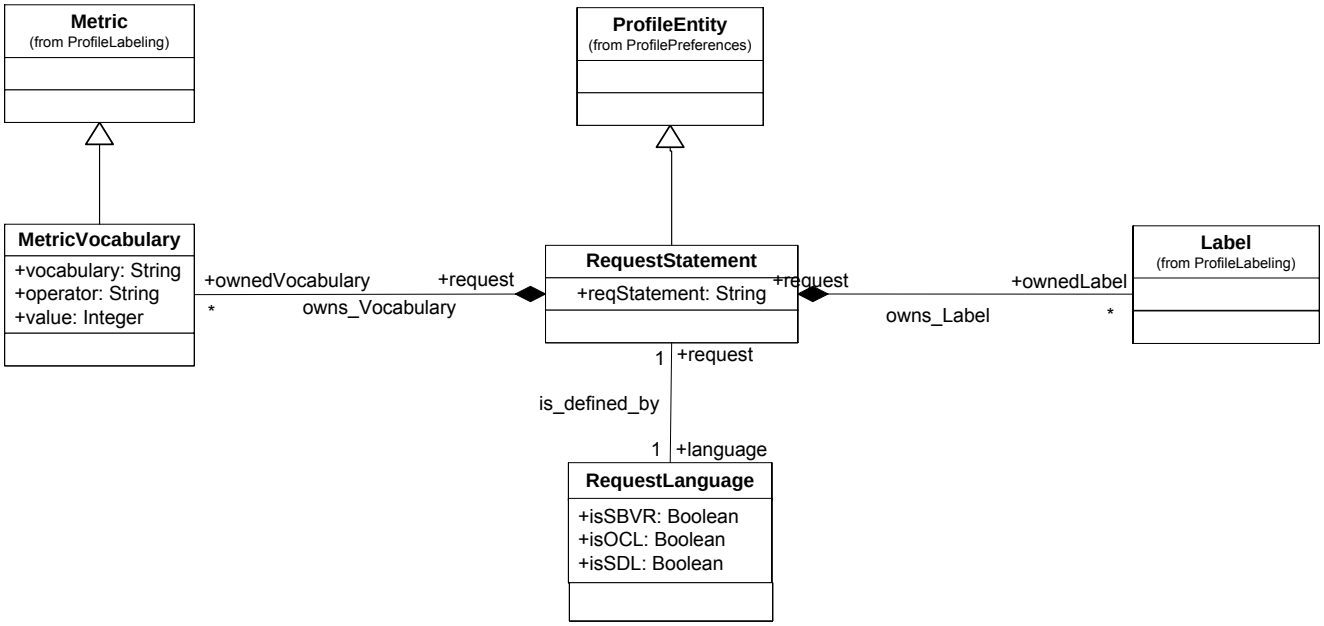


Figure 13: Profile Request Package

The class ***MetricVocabulary*** provides basic functionality for describing request statements using natural speech, numbers and operators.

Variable	Function	Example
vocabulary	All words which can be used for defining requests	Room, non-smoker, is, has, …
operator	Operators which can be used for defining constraints	<, >, >=, <=, !=, ->, …
value	The values within constrainst	200 €, 5 km, …

Table 7: Profile Request – Class MetricVocabulary

The class ***RequestLanguage*** adds information regarding the type of language used for describing requests:

Variable	Function	Example
isSBVR	the type of language used for requests	isSBVR = true
isOCL	the type of language used for requests	isOCL = false
isSDL	the type of language used for requests	isSDL = false

Table 8: Profile Request – Class RequestLanguage

The class ***RequestStatement*** represents the final request concatenated by the values from related classes:

Variable	Function	Example
reqStatement	the stored request statement	- “Room has Minibar” - ”RoomRate <= 200 €” - “context Room inv TargetRoomRate: RoomRate <= 200”

Table 9: Profile Request – Class RequestStatement

5 CONCLUSIONS

The deliverable on hand gives an introduction on how the architecture of a user profile could look like and how the handling of user profile information could be done. For this, information about the user and his existing basic IT infrastructure has been described. Additionally a set of exemplarily dimensions and metrics has been given. The idea of an extended BML gives the opportunity to represent additional information next to the business related ones. This could be for example the representation of technical information like data for Web Services and so on. The internal organisation is thought to be realised in a dynamic way using timestamps for data sets to keep a profile up to date and to eliminate possible dead data sets which doesn’t map a user’s preference anymore.

6 APPENDIX

6-1: User Information (Data Examples):

Asset	Attributes
Company	Name Legal Form Business Status
BusinessData	Number of Employees Total Revenue per year EBIT (Earnings Before Interests and Taxes) Net Profit
Location	Country County Language
Address	Street Nr. City Province Postal Code
Communication Channels	Telephone Mobile Fax Email URL
Contact	Name Position Department Authorities
Sector	SecName SecPath (general to specific branch path) Position in Supply Chain
Client Structure	Nationality Institution (Company, Organisation, Person) Pattern of Distribution Number of Clients Motivation (private, business)
Processes	TypeOfReservation (directly, via agency, via third party) MediumForReservation (Email, Telefone, SMS MMS, Fax etc.)
Services	Number of Outsourced Services Name of Outsourced Service Number of Offered Services Name of Offered Service Number of Requested Services Name of Requested Service

	Number of Rated Services Name of Rated Service
…	…

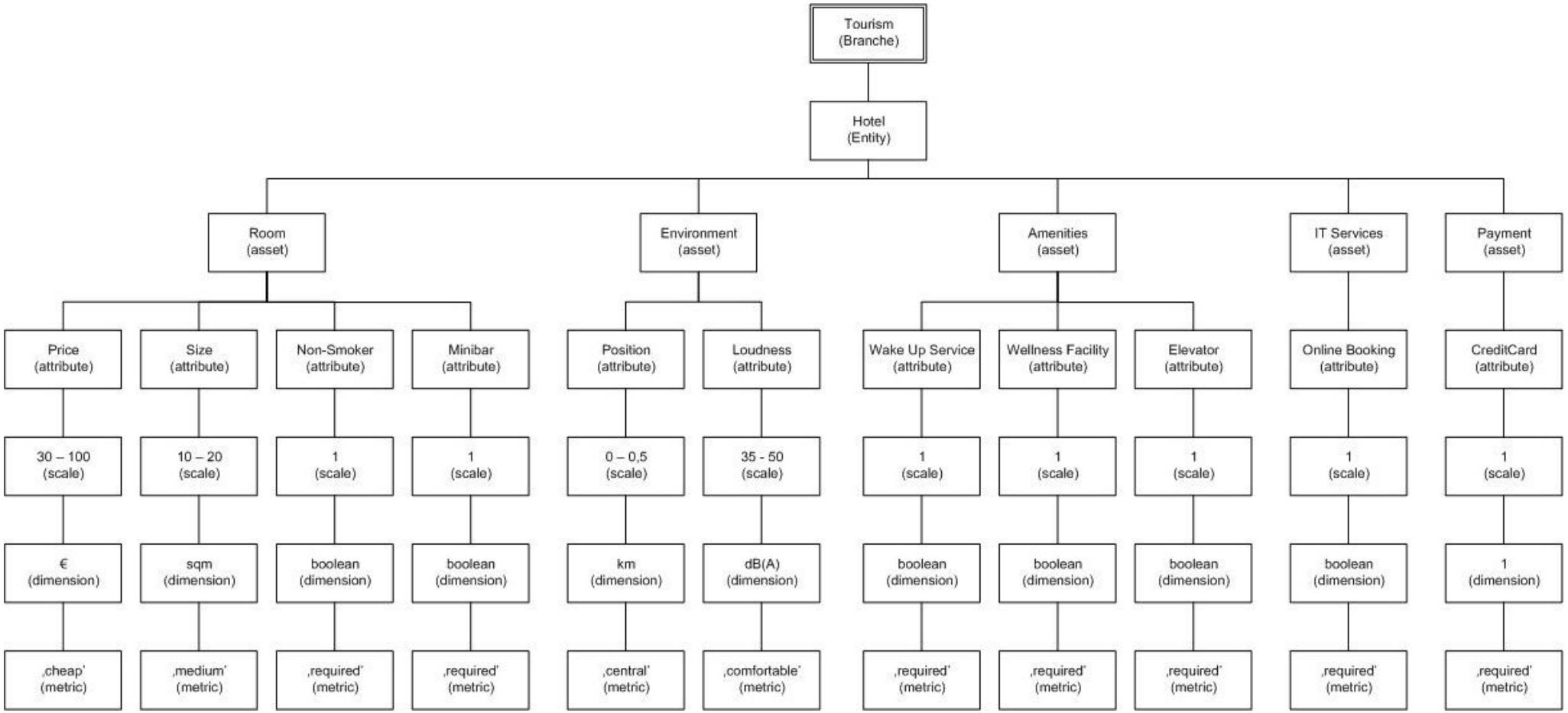
6-2: The Rathbone Code

Value Codes				
	M Materials			
		M1 Extraction		
		M2 Farming		
	C Conversion			
		C1 Continuous Process		
			C1(M) Manufacturer	
			C1(S) Contractor	
		C2 Batch Process		
			C2(M) Manufacturer	
			C2(S) Contractor	
		C3 Assembly		
			C3(M) Manufacturer	
			C3(S) Contractor	
		C4 Intellectual		
			C4(M) Manufacturer	
			C4(S) Contractor	
		C5 Craft		
			C5(M) Manufacturer	
			C5(S) Contractor	
	S Services			
		S1 Intellectual		
		S2 Craft		
		S3 Ressources		
	D Distribution			
		D1 Stockist		
			D1a Trade	
			D1b Retail	
		D2 Agent		

Growth Stage Codes			
	-1 Pre-Start Up		
		0 Start-Up	
		1 Combined Management and Workforce	
			1+ First Pain Barrier
		2 Separate Management	
			2+ Second Pain Barrier
		3 Multi-Layer Management Structure	
		9 No Internal Management	

Personality Codes				
	L	Private Lifestyle		
		L1	One Owner	
		L2	Multi-Ownership	
		L3	Family Business	
		L4	Subsidiary	
			L4(D)	Dependent (or Inter-Dependent)
			L4(I)	Independent
	H	Private High Growth		
		H1	One Owner	
		H2	Multi-Ownership	
		H3	Family Business	
		H4	Subsidiary	
			H4(D)	Dependent (or Inter-Dependent)
			H4(I)	Independent
	P	Public		
		P1	Immature	
		P2	Mature	
	Z	Restructure (Flotation / Liquidation / Buy-Out)		

6-3: Profile Data Structure Example



7 REFERENCES

[Andr98a] Andrews, J. H.: Theory and practice of log file analysis. Technical Report 524, Department of Computer Science, University of Western Ontario, 1998.

[Andr98b] Andrews. - J. H.: Testing using logfile analysis: tools, methods, and issues. Proc. 13th IEEE International Conference on Automated Software Engineering, 1998, pp. 157-166.

[Dych02] Dyche, J.: The CRM Handbook – A Business Guide to Customer Relationship Management. 2nd Printing, Addison-Wesley, Indianapolis, 2002.

[EvCe04] The Evaluation Center: Teacher Evaluation Kit - Complete Glossary. http://www.wmich.edu/evalctr/ess/glossary/glossary.htm. Request on 2005-04-19.

[IEEE90] IEEE: IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610, Piscataway, New Jersey, 1990.

[ISUF04] Istituto Superiore Universitario Formazione Interdisciplinare: BML Metamodell – Report Draft (Task B18). https://bml.digital-ecosystem.net/files/documents/13/453/BML_metamodel_2.0.1.zip, Request on 2005-03-03.

[Klüv01] Klüver, W.: Software Engineering – Kalkulation. http://www.fh-augsburg.de/informatik/vorlesungen/se1t/script/management/direkt.html. Request on 2005-03-23.

[Kova05] Kovacs, J.: Building A Business Relationship. http://ezinearticles.com/?Building-A- Business-Relationship&id=10746, Request on 2005-02-16.

[Kowa99] Kowalski, G.: Information Retrieval Systems – Theory and Implementation. 3rd Printing, Kluwer Academic Publishers, Massachusetts, 1999.

[OeFA00] Österle, H.; Fleisch, E.; Alt, R.: Business Networking – Shaping Enterprise Relationships on the Internet. Springer-Verlag, Berlin-Heidelberg, 2000.

[Rath04a] Rathbone, N.: Background to the SME coding system. 2004.

[Rath04b] Rathbone, N.: Understanding the DBE codes. https://training.digital-ecosystem.net/files/documents/22/554/Understanding_the_DBE_codes.ppt. Request on 2005-04-23.

[Solu04] Soluta.Net: DBE Architecture Requirements – Preliminary Design, Usage Scenarios and Architecture Requirements. https://dev.digital-ecosystem.net/files/documents/20/305/DBEArchitectureRequirements.pdf. Request on 2005-05-07.

[TDM04] Target Domain Meeting 2004-04-26: Summarised and extended Minutes. https://regions.digital-ecosystem.net/files/documents/26/132/Target DomainsMinutes.doc. Request on 2005-01-17.

[Vald01] Valdman, J.: Log File Analysis - PhD Report. http://www.kiv.zcu.cz/ publications/2001/tr-2001-04.pdf. Request on 2005-04-01.

[W3C03] W3C Working Group Note: QoS for Web Services - Requirements and Pos-Sible Approaches. http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/. Request on 2005-03-08.

[WIT04] Waterfold Institute for Technology: State of the Art in Accounting for Composed Services (Deliverable 36.1). https://dev.digital-ecosystem.net/ files/documents/20/292/D36-1-Final.pdf. Request on 2005-02-25.