Sales Forecasting as a Service
A Cloud based Pluggable E-Commerce Data Analytics Service

Fabian Aulkemeier[1], Roman Daukuls[2], Maria-Eugenia Iacob[1], Jaap Boter[2], Jos van Hillegersberg[1] and Sander de Leeuw[2,3]
[1]Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands
[2]Faculty of Economics & Business Administration, VU University, Amsterdam, The Netherlands
[3]Nottingham Business School, Nottingham Trent University, Nottingham, U.K.

Keywords: Data Analytics, Sales Forecasting, Cloud Service, Platform Architecture.

Abstract: Data analysts are increasingly important for companies to extract critical information from their vast amount of data in order to be competitive. Data analytics specialists or data scientists develop statistical models and make use of dedicated software components for example to categorize products and forecast future sales. Their unique skill set is among the most sought after in the current job market. Cloud computing on the other hand helps companies to acquire services in the cloud and share the required expertise for delivery among service users. In this paper we take a cross disciplinary approach to develop a data analytics technique and a platform based IT architecture that allows to outsource sales forecasting analytics into the cloud.

1 INTRODUCTION

In this information age, the expertise of data analytics specialists or data scientists has become a critical success factor for organizations to understand and react to their environment. The shortage in skilled professionals and the resulting high cost causes a deficit of such experts in many domains (Davenport and Patil, 2012). As a consequence, notably small and medium enterprises (SME) often miss the potential that lies in unexploited information.

In the domain of e-commerce, sales forecasts provide an example of such critical information. Online retailers that are able to compute reliable forecasts, based on existing sales transactions can reduce losses caused by out of stock or non-selling items. Especially in short series product life cycle fields such as fashion, it is crucial to have accurate figures on upcoming sales even before production.

Among SMEs cloud computing in general and software as a service (SaaS) in particular are popular solutions to share the costs for IT service development and operation (Danaiata and Hurbean 2010). Therefore, the task of data analytics for product sales forecasting is a promising application for the new cloud service model.

With the current system landscape of most online retailers, transactional data is scattered across various application system components and has to be preprocessed before it may be used. Data preprocessing consists of data cleaning, record selection, summarization, denormalization, variable creation and coding. It is considered as the most time-consuming task in data analytics projects (Ordonez 2011). However, collecting and cleansing data from various sources is a very customer specific task and therefore difficult to implement as a cloud service.

The CATeLOG project which is financed by Dutch institute for advanced logistics (Dinalog) aims amongst others at the development of innovative, pluggable e-commerce services as well as a suitable platform architecture to facilitate the adoption of such services. For this work we have combined two areas of expertise within our research project to come up with a solution to develop state of the art sales forecasting logic and integrate it into a pluggable platform architecture. The research goal was to design and develop a cloud based sales forecasting service to allow small and medium enterprises to make use of advances in data analytics techniques.

In section 2 we present the current research in sales forecasting and present a forecasting module which is the core component of the solution. In the third section we outline the concept of service pluggability and present the architecture for a pluggable service platform. In section 4 we present

Aulkemeier, F., Daukuls, R., Iacob, M-E., Boter, J., Hillegersberg, J. and Leeuw, S.
Sales Forecasting as a Service - A Cloud based Pluggable E-Commerce Data Analytics Service.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 2*, pages 345-352
ISBN: 978-989-758-187-8

the prototype of a cloud based forecasting service and evaluate its pluggability.

2 NEW SALES FORECASTING MODULE

New product sales forecasts are valuable for managers in supporting important decisions in operations planning (Cohen et al. 2000). For example, managers need to know how sales will evolve in the future in order to determine purchasing quantities and inventory. The number of new product introductions has been increasing over the past decade. As a consequence, managers need to perform new product sales forecasting tasks more frequently than in the past.

Despite the importance and prevalence of new product forecasts, these forecasts are seldom accurate. Kahn (2002) found that new product forecast accuracy was 58%, based on interviews with managers. Moreover, there are numerous approaches to forecast new product sales, and these approaches may perform differently under different circumstances. Hence, managers would benefit from a tool that can incorporate multiple approaches to forecast new product sales and pick the most accurate approach.

In what follows, we briefly discuss new product forecasting approaches and describe the development and implementation of the new product sales forecasting module.

2.1 New Product Sales Forecasting Approaches

Goodwin et al. (2014) distinguish three categories of new product forecasting approaches: managerial judgement, judgement by potential customers and formal models. Managerial judgement relies on managers providing estimates 1of sales expected, typically using experience. Judgement by potential customers may involve for example expert panels. Formal models make quantitative projections using mathematical formulations of relationships between relevant variables. Since our goal is to provide a forecasting tool that is modular and can be used by multiple clients, we focus on formal models.

There are numerous formal models that can be used to forecast new product sales. These formal models come from both Statistics and Machine Learning areas. Some models have roots in marketing or economic theory, while others are purely data-driven. We will use a commonly used statistical model based on Marketing theory (the Bass model), and one more data-driven model (latent-class regression).

2.1.1 Bass Model

Originally proposed by Bass (2004), the Bass model and its derivatives are widely used in Marketing to characterize the life cycle of a product. It has the following formulation:

$$\frac{f(t)}{1-F(t)} = p + qF(t) \quad (1)$$

where $f(t)$ is the probability density function of adoption time t and $F(t)$ is the cumulative distribution function of t. The model postulate that the hazard of adoption depends on two forces: external influence p and internal influence $qF(t)$. The model can be rewritten in the cumulative sales domain as:

$$S_i(t) = m_i p_i + (q_i - p_i)CS_i(t) - \frac{q_i}{m_i}[CS_i(t)]^2 \quad (2)$$

where $S_i(t)$ i4s new product sales of product i at time t, $CS_i(t)$ is cumulative sales of product i at time t, and m_i is the market potential. This equation can be estimated on historical data using ordinary least squares, and parameter estimates for p_i, q_i and m_i can be obtained.

In order to forecast new product sales before launch, the parameters of the Bass curve for the new product can be estimated using analogy by considering “similar” products introduced in the past (Bass 2004).

2.1.2 Latent-class Regression Model

We furthermore consider a latent-class Poisson regression model (Wedel et al. 1993) in combination with concomitant variables (Grün and Leisch 2008). The advantage of this model is that it estimates the two models simultaneously: one model for clustering product life cycles and one model for assigning cluster probabilities to each instance based on concomitant variables. The formulation of this model is as follows:

$$h(S_{it}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{X},\boldsymbol{Y}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{\alpha},\boldsymbol{Y}) P_{it|k}(S_{it}|\boldsymbol{\beta_k},\boldsymbol{X}) \quad (3)$$

where h denotes the mixture density, k is the class index (K classes in total), S_{it} is sales of product i at time t, $\boldsymbol{X}$ is the matrix of independent variables influencing the product life cycle pattern, $\boldsymbol{Y}$ is the matrix of independent variables influencing cluster membership (concomitant variables); $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and π_k are parameters to be estimated.

Further, we assume that the number of products sold follows the Poisson distribution with rate parameter that depends on $\boldsymbol{\beta_k}$ and $\boldsymbol{X}$ in each cluster.

2.2 Algorithms and Implementation

We build our forecasting module using the R software (R Development Core Team 2014). It has a number of advantages, including the fact that it is free, open-source, and contains a large number of statistical packages that facilitate rapid model development.

Before the sales forecasting module can be utilized, it is very important to supply clean data to the module in the format that it requires. In our application (cf. 2.4), data had to be aggregated, cleaned, and new features had to be derived. Some of these procedures require domain-specific knowledge. It can therefore be expected that data preprocessing may encompass different procedures for different clients. R data management libraries, such as plyr (Wickham 2011) or caret (Kuhn and Johnson 2013), may greatly facilitate the development of client-specific data handling procedures.

There are three key functions in the sales forecasting module: (1) model tuning, (2) best model selection and (3) forecasting. Model tuning refers to finding the best model-specific setting. For example, in the case of latent-class regression, model tuning implies selecting the number of clusters that minimizes cross-validation error. Selecting among models is done based on cross-validation errors – we select the model with the least error. This model is used to perform the forecasting task. For more details on model tuning and selection, we refer the reader to Kuhn and Johnson (2013).

Both forecasting models described in section 2.1 were implemented in R using the stats (R Development Core Team 2014) and the flexmix (Grün and Leisch 2008) packages.

2.3 Case Description

We demonstrate the functionality and performance of our forecasting module by using sales data of a large Dutch apparel retailer. In apparel retailing, assortments are renewed at least two times per year, and new item introductions are common.

Our data includes monthly sales of 43 brands in the period between 05-02-2009 and 21-02-2013. A brand can have multiple collections, styles, colours and sizes. We aggregate across these variables to arrive at brand sales data. We observe average prices, discounts, inventory levels and number of unique stock-keeping units within each brand. Each brand is characterized by its functionality (an internal company classification variable) and average non-discounted price. Sales series of four selected brands are shown in Figure 1.

We can see that brands exhibit different life cycle patterns and that sales peak at different moments.

2.4 Capabilities and Output of the New Product Forecasting Module

Our objective is to forecast sales of new brands in a given category prior to their launch. We split the data into a training set and a test set, where training set data include observations until 01-01-2011, and the test set

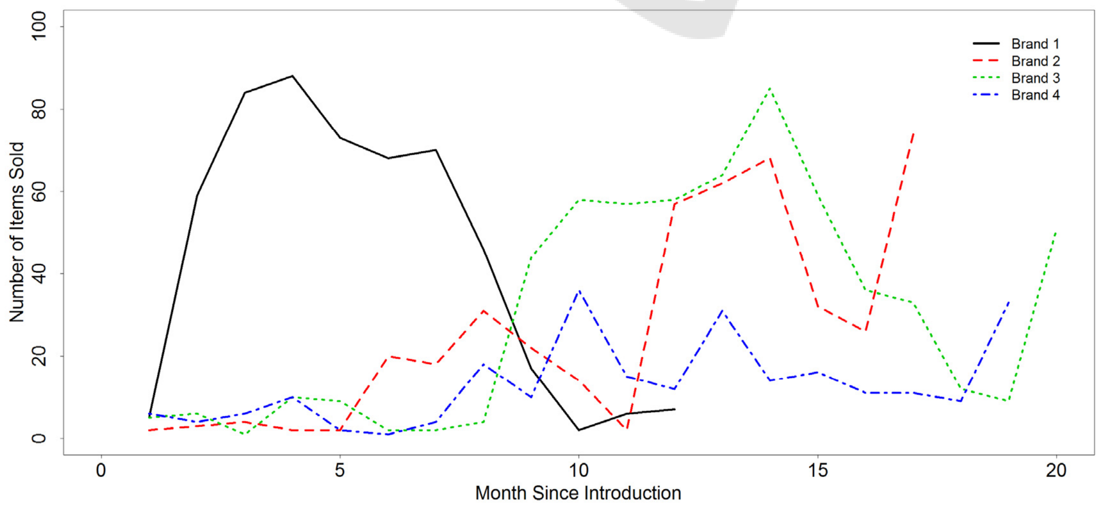

Figure 1: Sales series of selected brands.

includes sales data on new products that are launched after this date. The splitting procedure resulted in 22 brands in the training set and 21 brands in the test set. We tune the models and select the best-performing model on the training set, and evaluate forecasting performance on the test set. Therefore, we imitate a real-life scenario of a pre-launch forecast.

2.4.1 Training Set Results

We used leave-one-out cross-validation to tune and evaluate the Bass model and the latent class regression. For the Bass model, we first fit the Bass curve to each brand separately and obtain the parameters. Next, to predict the sales curve of a new brand, we use its attributes to compute the "distance" between the current brand and all brands in the training set. We use n closest brands and computed the average p, q and m values. Next, based on these average values, we predict the sales of the new brand. Hence, n, the number of closest brands to consider, is our tuning parameter. For the latent-class regression, we used leave-one-out cross-validation to fit the model with k different clusters. Hence, k is the tuning parameter for the latent-class regression model. The optimal configuration, giving the lowest mean absolute percentage error (MAPE) turned out to be $n = 2$ and $k = 3$.

We compare the MAPEs of both models with the best configurations using the t-test. We reject the null hypothesis that the mean performance of the Bass model is better than that of latent-class model with p-value of 0.01. Thus, we expect the latent-class model to perform better on the test set.

2.4.2 Forecasting Performance

We use both models to predict the sales of new products in the test set. Table 1 provides MAPEs, aggregated across brands, for each month since new brand introduction. It is important to note that the results described in this section are preliminary and should be interpreted with care.

We can see that the latent class regression performs better than the Bass model. This is due to the fact that it incorporates decision variables (pricing, discounts and stock levels). The forecast errors are rather high for both models. This is

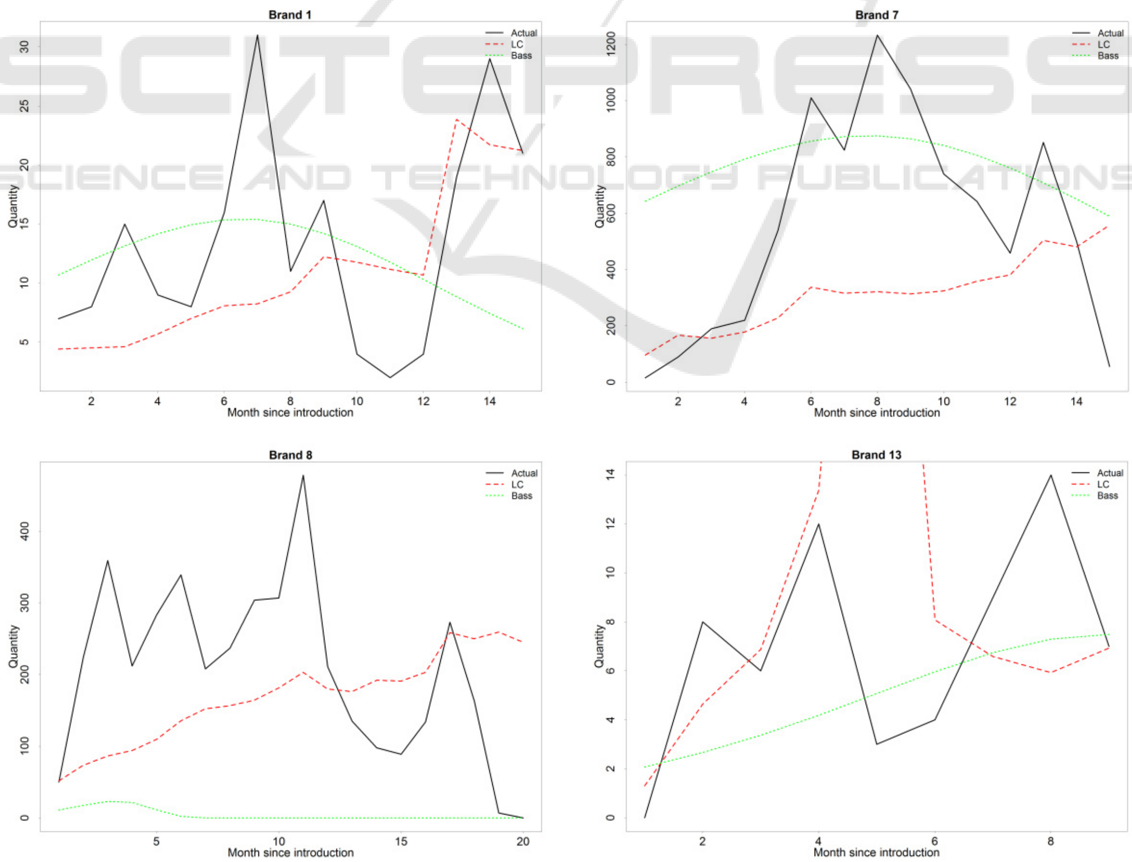

Figure 2: Forecast vs. actual sales for selected items.

Table 1: Performance on test set.

Months since introduction	MAPE Bass model	MAPE Latent class regression
1	1119	158
2	1036	115
3	3886	156
4	2037	99
5	1005	78
6	2405	100
7	4751	261
8	900	157
9	2122	103
10	1695	131
11	360	153
12	449	217

possibly due to the fact that we do not have data on many brand attributes, and it is difficult to establish similarity between brands based on current attributes alone. Figure 2 provides several plots with actual and forecast sales for several brands.

3 PLUGGABLE ARCHITECTURES

In section 2 we have shown how past sales transactions can help retailers to predict future sales. However, to obtain a complete, ready to use cloud based forecasting solution, architectural questions arise and will be discussed in the following. First, we introduce the concept of pluggability, which will help us to understand the issues of the forecasting module and later on, to evaluate the solution proposed in this paper. Secondly, we present a platform architecture which forms the base of the implemented prototype in section 4.

3.1 Pluggability

The requirements for software components usually go beyond their pure functionality. Such non-functional requirements are also known as software quality characteristics. While the practice of software quality goes back to the 70s (McCall et al. 1977), specific quality characteristics for service-oriented systems have evolved recently and mostly aim at a higher agility of the resulting system (Lankhorst et al. 2012). For cloud services, pluggability is one of such quality characteristics. It focuses on limiting the efforts of adopting new services (Aulkemeier et al. 2015). In order to transform the core forecasting module into a pluggable cloud service, six criteria have to be met. In the following we are discussing the forecasting module with regards to the criteria.

Ease of Provisioning (EOP) is the ability of the service to support the user in selecting a suitable service and to anticipate the costs, efforts and benefits of its use. In case of the forecasting module, it is not possible to oversee the capabilities of the module from a business user perspective. Thus, it is difficult to predict the costs, associated with the transformation of the module into a ready to use business application.

Ease of Deployment (EOD) means to minimize the efforts for installing the service, including the allocation of hardware and system software resources. If the forecasting module would be distributed as is, the user would have to deploy suitable hardware and software as well as to provide suitable application components in order to support the end user.

Ease of Adaptation (EOA) has two different aspects. The adaptation through configuration by a business user as well as the adaptation of the service by technical experts. The forecasting service offers a maximum flexibility for software developers to reuse the forecasting functionality. However, it does not support a configuration by a non-technical business user.

Ease of Integration (EOI) describes the capability of a service to interact with other IT components. The data gathering and preprocessing tasks mentioned earlier can be considered as aspects of integration. The forecasting module requires the preprocessed data as input and does not support the user with gathering and cleansing the data from other services.

Ease of Operation (EOO) encompasses all continuous tasks after setting up the service. Maintenance of services includes, for example, bug fixing, functional enhancements, security updates, and end user support. While bug fixes and enhancements could be distributed in an automatic fashion, the maintenance of potential additional application components and end user support needs to be carried out by the consumer.

Ease of Exchange (EOE) of a service often relates to the dependencies with other components. If *services* depend on each other, the process of exchange is getting more complex. Services such as the forecasting module that only serve reporting purposes can usually be removed without affecting the rest of the landscape.

It is clear that the forecasting module described above does not cover all quality criteria of a pluggable service. Thus, in the following section, we present an architecture for a pluggable forecasting service with the module at its core.

3.2 Architecture

In order to support retailers with the adoption of innovative e-commerce services we created the CATeLOG platform. At its core the platform contains a canonical data model (CDM) to share e-commerce related information across services. Furthermore, it provides an application programming interface (API) to give service providers access to the shared resources. It allows the platform clients to implement e-commerce services in a federated fashion (Busse et al. 1999). The CDM and the federated nature of the platform help to reduce the efforts in data gathering and preprocessing required for the forecasting service.

The architecture of the forecasting service and the interaction with the platform is shown in Figure 3. The forecasting service component has four application functions. It interacts with the platform through the same API as order management, online store, product information management and other services. It provides a wrapper around the core forecasting module, which transforms the data from the platform into the suitable format, triggers the prediction module generation and requests individual forecasts. The subscription mechanism uses the standard authentication flow provided by the CATeLOG platform. It allows potential users to subscribe to the service by entering its platform credentials and granting the service access to the shared resources. The web application allows the user to configure the service and displays the output of the forecasts. Finally the scheduler is available for long running jobs. As the generation of the prediction models usually takes a longer time it has to be done in the background. Users can schedule the model generation periodically or on demand.

4 A PLUGGABLE SALES FORECASTING SERVICE

In order to evaluate the architecture we created a prototype of the platform as well as the forecasting service. In the following we provide a description of the prototype and evaluate the pluggability of the implemented forecasting service.

4.1 Prototype

A prototype has been developed using standard web application technologies, an SQL database for storing data within and outside the scope of the platform as well as various common libraries for web APIs, OAuth authentication flow between the platform and the service, interfacing the R module, and job scheduling.

Figure 4 shows the web application user interface. The user has the option to choose between various forecasts and to schedule the prediction model and forecast generation jobs.

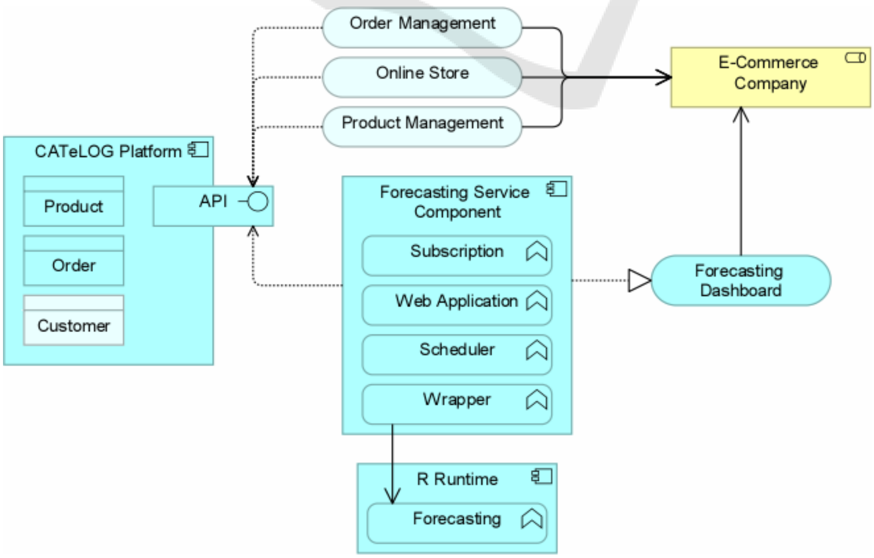

Figure 3: Forecasting service architecture.

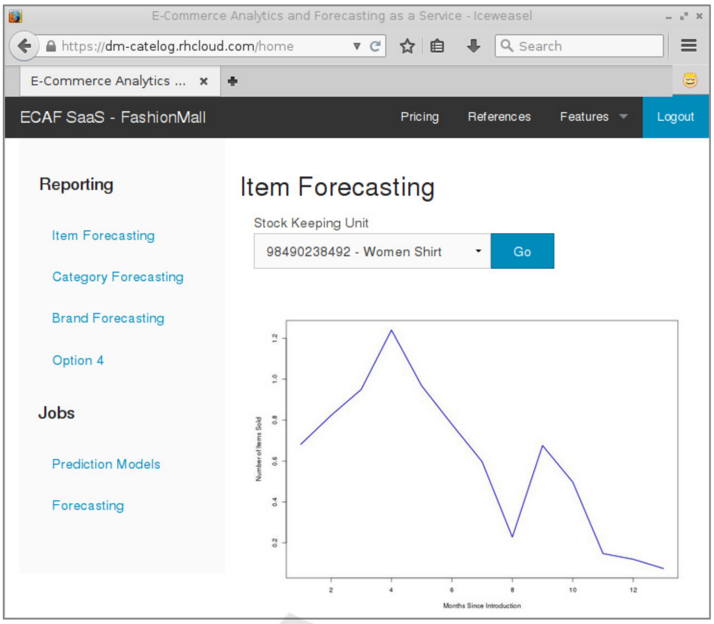

Figure 4: "e-Commerce analytics and forecasting SaaS" web application.

4.2 Evaluation

The goal of the architectural design and prototype was to transform the state of the art forecasting module into a pluggable cloud service. According to prevailing design science research methodologies (Peffers et al. 2007) the design and demonstration of a design artefact should be evaluated by observing if the artefact provides a solution to the design goals. In the following, we do so by comparing the pluggability of the implemented service with the pluggability of the forecasting module in section 3.1.

- EOP: By introducing the service with the forecasting *module* at its core, we can achieve a higher abstraction of service. While the forecasting module is providing functionality, the service is providing business value (Haesen et al. 2008). Thus, for the potential user, it gets easier to map the service features to the business requirements, eventually improving the EOP. Furthermore, if the service is offered as a platform based artefact, it is possible to discover and compare the services in a marketplace fashion. This could further improve the EOP of the service over the plain module.
- EOD: As the forecasting service is cloud based the user does not need to deploy any software or hardware. The service offers a subscription by using the platform credentials. Directly after subscribing to the service, the user can go over to the adaptation phase.
- EOA: Similar to the aspects of the provisioning phase, the service also supports the user in terms of service adaptation. The configuration and setup can be done within the web interface, resulting in a higher EOA. However, the possibilities of service customization through developers are very limited, resulting in less flexibility in adaptation. This is a common disadvantage of cloud services compared to custom or packaged solutions.
- EOI: The EOI depends on the work that is necessary to connect the service with other components. In order to operate accurately, the forecasting module requires a preprocessed data set that contains the right data fields and records. In current e-commerce architectures, sales transactions and product information are stored across different systems such as product information management, order management or online shop frontend. By relying on the platform architecture the service provider can pre-integrate the service with the CDM. Thus, the service user does not have any additional tasks with regards to service integration and the EOI could be improved significantly by using the platform.
- EOO: By making the service cloud based, the service operation is shifted from the user to the service provider. The EOO is higher as the user does not have to carry out any maintenance tasks.
- EOE: As mentioned earlier the EOE has limited relevance for reporting services. However, the shared backend of all services in the platform guarantees that exchanging the service may not affect the integration points with other components.

5 CONCLUSIONS

In the previous sections we have shown how a cloud based forecasting service can be designed and implemented based on a state of the art forecasting module. Furthermore, we verified the pluggability of the prototype with regards to the six criteria. It was shown that the pluggability of the service exceeds the pluggability of the plain forecasting module, and offers the user a solution that is easy to adopt. The solution can be particularly interesting for SMEs that do not have the resources for a comparable on premise solution. However, it is required that the platform is in place and an ecosystem of services and service providers has been established.

In this paper we have only given a short description of the CATeLOG platform as the focus of this work was on the transformation of the forecasting module into a pluggable service. In parallel and future publications we concentrate on the architecture of the platform, its functional requirements, and further benefits.

REFERENCES

Aulkemeier, F., Iacob, M.-E. & Hillegersberg, J. van, 2015. Pluggable SaaS Integration: Quality Characteristics for Cloud Based Application Services. In *Enterprise Systems Conference (ES)*. Basel.

Bass, F. M., 2004. Comments on “A New Product Growth for Model Consumer Durables The Bass Model.” *Management Science*, 50(12_supplement), pp.1833–1840.

Busse, S. et al., 1999. *Federated information systems: Concepts, terminology and architectures*, Technische Universität Berlin, Fachbereich 13-Informatik.

Cohen, M. A., Ho, T. H. & Matsuo, H., 2000. Operations Planning in the Presence of Innovation-Diffusion Dynamics. *New-Product Diffusion Models*.

Danaiata, D. & Hurbean, C., 2010. SaaS–Better solution for small and medium-sized enterprises. In 2nd Multiconference on Applied Economics, Business and Development (AEBD ’10). Kantaoui, Sousse, Tunisia.

Davenport, T. H. & Patil, D.J., 2012. Data Scientist: The Sexiest Job of the 21st Century. *Harvard Business Review*, p.70.

Goodwin, P., Meeran, S. & Dyussekeneva, K., 2014. The challenges of pre-launch forecasting of adoption time series for new durable products. *International Journal of Forecasting*, 30(4), pp.1082–1097.

Grün, B. & Leisch, F., 2008. FlexMix version 2: finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4), pp.1–35.

Haesen, R. et al., 2008. On the Definition of Service Granularity and Its Architectural Impact. In Z. Bellahsène & M. Léonard, eds. *Advanced Information Systems Engineering*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 375–389.

Kahn, K. B., 2002. An exploratory Investigation of new product forecasting practices. *Journal of Product Innovation Management*, 19(2), pp.133–143.

Kuhn, M. & Johnson, K., 2013. *Applied predictive modeling*, New York: Springer.

Lankhorst, M. M. et al., 2012. Agility. In M. Lankhorst, ed. *Agile Service Development*. The Enterprise Engineering Series. Springer Berlin Heidelberg, pp. 17–40.

McCall, J. A., Richards, P. K. & Walters, G. F., 1977. *Factors in software quality. Volume I. Concepts and Definitions of Software Quality*, Sunnyvale, CA: General Eletctric Company.

Ordonez, C., 2011. Data Set Preprocessing and Transformation in a Database System. *Intell. Data Anal.*, 15(4), pp.613–631.

Peffers, K. et al., 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), pp.45–77.

R Development Core Team, 2014. *R: A language and environment for statistical computing* the R Foundation for Statistical Computing., Vienna, Austria.

Wedel, M. et al., 1993. A latent class poisson regression model for heterogeneous count data. *Journal of Applied Econometrics*, 8(4), pp.397–411.

Wickham, H., 2011. The Split-Apply-Combine Strategy for Data Analysis. *Journal of Statistical Software*, 40(1), pp.1–29.