Controlling Open Source Intermediaries – a Web Log Mining Approach

Heinz Lothar Grob, Frank Bensberg
Department of Information Systems, University of Muenster
Leonardo Campus 3, D–48149 Muenster, Germany
grob@uni-muenster.de, bensberg@uni-muenster.de

Firoz Kaderali
Department of Communication Systems, University of Hagen
Universitätsstrasse 11, D–58084 Hagen, Germany
firoz.kaderali@fernuni-hagen.de

Abstract. *Open Source Software (OSS) has become a focal research issue in computer science. One interesting phenomenon in the OSS market is the evolution of dedicated intermediaries, which provide OSS-related services to the OSS community. Prevalently, OSS intermediaries fail to fulfil their financial and non-financial goals, such that many of them have vanished from the OSS market or significantly changed their business models. A major reason for this development is the lack of appropriate instruments which are able to dismantle coordination problems and failures inherent to the OSS development model.*

This paper proposes a controlling instrument for OSS intermediaries based on a behaviouristic approach. Since OSS intermediaries typically provide internet based services, it is possible to observe actor behaviour by use of technical data acquisition instruments like server logging. Therefore, it seems reasonable to examine the concept of web log mining, which provides a framework to analyse server based log files. Finally, we apply this concept to the research field of OSS intermediaries and show empirical results. These results relate to the OSS intermediary CampusSource, who provides e-learning software and complementary services.

Keywords. Open Source Software, CampusSource, E-Learning, Intermediary, Software Market, Controlling, Web Log Mining, Data Mining, Knowledge Discovery.

1. Introduction

In 2002, Europe represented 29 % of the global market for information and communication technology (ICT). Within this sector, the software market has only gained marginal importance compared to other segments like hardware or services. In numbers, the software market only amounts to 10 % of the total ICT sector [16]. The reasons for this shortcoming of the European software industry are partially caused by dependencies on foreign mono- and oligopolies, which lead to fragmented software markets and impose significant constraints for market entry, in particular for innovative SMEs. Since software production is the main driver for complementary services and consulting, the whole ICT market lags behind its original potential.

A promising approach to stimulate collaboration along with incremental and complementary innovations in domestic software markets and local industries is the promotion and employment of Open Source Software (OSS). OSS has important implications for the European software industry because it represents a body of emerging competence and specialisation that can provide the basis for system consulting and further software development efforts. Practical experience reveals that OSS leads to greater independence and increased sustainability of software investments, as well as better cooperation between research and economy.

While the continuing increase in the significance of OSS has led to a situation where a large number of software components are made available under open source licence conditions, the use of open source software components in the industry is still at a relatively low level [7]. The reasons for this low diffusion of OSS lie in the technological, economical and legal barriers inherent in the supply and demand sides of the OSS market. Existing OSS intermediaries seem to be a first step to dismantle these market-dominating barriers. For example, the OSS intermediary SourceForge.net hosts more than 70,000 open source projects, which represent a substantial economic potential for the European software industry. Through its internet commu-

nity platform, SourceForge.net supports services for OSS development like project management, mailing lists, version management and bug tracking.

Most OSS intermediaries exhibit business models and provide complementary services for a fee. However, practical experience reveals that many OSS intermediaries fail to fulfil their business objectives, predominantly because of their inability to match the preferences of the selling market [15]. Therefore, it is necessary to devise adequate controlling instruments, which break down this coordination deficiency.

From a research perspective, it seems interesting that – according to the best of our knowledge – yet no scientific effort has been made to tackle the field of controlling instruments for the domain of OSS intermediaries. A reason for this deficit is that the theory of management and controlling primarily focuses on traditional enterprises, whereas OSS intermediaries imply a set of peculiarities, which prevent a transfer of established controlling instruments. For instance, these peculiarities are:

- the intangibility of OSS as incremental and complementary innovation,
- the distributed, networked and collaborative open source software production processes, and
- the technically mediated interactions between supply and demand side by continuous use of the internet.

Consequently, controlling instruments have to address these predominant traits of OSS intermediaries. A major analytical task of such instruments is to identify preferences and interests of the demand side. This provides relevant management information that can be used to allocate open source development effort according to market requirements and therefore prevents coordination deficiencies between the supply and the demand side. For sure, this analytical task requires adequate data acquisition methods. Due to the fact that the interactions of intermediaries with the demand side are organised via internet applications (e. g. centralised portals like SourceForge.net or CampusSource.de/org/), online behaviour can be explicitly observed by technical means, in particular by server logging.

In order to analyse log file data, the concept of web log mining is applied. This provides an analytical framework for extracting relevant information from server based log files and is successfully used in the domain of e-commerce (e. g. online retailing, see [4]) to control online processes. Though, no scientific effort has been made in order to adapt web log mining to the domain of OSS intermediaries. Since failure rates of OSS intermediaries are high, this approach seems essential to resolve coordination deficiencies.

To create an appropriate controlling instrument, this paper discusses basic characteristics and functions of OSS intermediaries in a first step. Afterwards, the analytical framework of web log mining and the characteristics of its data source are introduced. Finally, we adapt this concept to the domain of OSS intermediaries and show empirical results for CampusSource, an intermediary for the domain of e-learning tools. These results highlight the informational potential of web log mining as a controlling instrument and serve to draw final conclusions.

2. Open source software and intermediaries

2.1 Characteristics of OSS

Open Source has become a focal research field, not only in computer science, but also in economics and jurisprudence. This interdisciplinary nature has led to a multitude of perspectives and research approaches to the open source phenomenon. Since we focus on economical implications of OSS, it is necessary to clarify material and immaterial product attributes. Therefore, we suggest a nested approach which integrates all disciplinary perspectives (fig. 1).

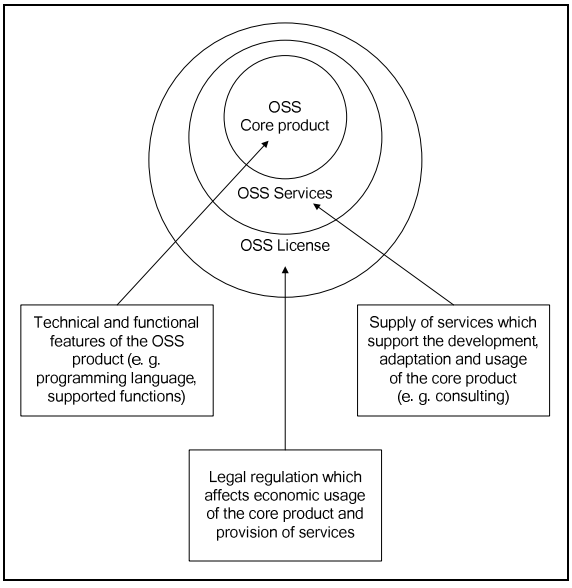

Fig. 1. Product attributes of OSS

From a technical perspective, an OSS product consists of a core product, which has basic technical and functional features. For instance, the open source web server *apache* provides the basic function to service requests via the hypertext transfer protocol (http) and is available for a vast multitude of software platforms. The next perspective comprises complementary services, which support the adoption process. For the product *apache,* services like installation, configuration and integration are available. Finally, OSS products possess specific legal features which are constituted in a license as a regulatory framework. This license authorises to perform specific actions with the product which are codified by copyright law. Usually, these comprise the reproduction, distribution and development of derivative products.

Commonly, Open Source Software is software released under a license conforming to the Open Source Definition (OSD) articulated by the Open Source Initiative [20]. According to the OSD, software has to comply to following primary legal criteria in order to be open source [6]:

- the complete source code must be available, e. g. via internet download,
- the user is entitled to redistribute the software for free or for fee,
- the software can be modified in order to create derived works, which in turn may be distributed, and
- the license must apply to all parties and must not exclude certain persons or groups.

Licensing is one obvious area of difference between open source and proprietary software products. The salient point is that OSS licensing not only entitles to *use* the software, but also to *modify* and *redistribute* the derived product (see [21], p. 311). This adds weight to the economical relevance of OSS as factor of production in the software market. Consequently, the liberal licensing of OSS has coined the development process, which differs from that of traditional commercial software (see [12], p. 48). Frequently, OSS development is carried out by a small group of authors organised within a loosely coupled project network, which is typically distributed (see [21], p. 312). Therefore, there is no "single point of control" as in hierarchically organised software production contexts in companies. Due to network coordination problems (see [24], pp. 190-192), open source software quality often does not match the requirements of the market. Examples for objective quality defects are missing functionalities and missing support services, consulting and documentation. In addition, also subjective quality characteristics affect the OSS adoption process, e. g. the absence of a reliable product roadmap which hedges system specific investments.

In order to dismantle these coordination problems between supply and demand side of the market, OSS needs an infrastructure to coordinate projects and resulting products. This is the scope of OSS intermediaries, which emerged during the last years.

2.2 Functions of OSS intermediaries

Research on intermediaries is predominantly pushed forward in the domain of electronic markets [18]. Theoretically speaking, the function of intermediaries is to coordinate transaction processes between the supply and demand side of a market (see [10], p. 407). Within the OSS market, intermediaries have to connect producers to consumers to promote market transactions. Supporting functions are for instance:

- development support services for open source projects, e. g. mailing lists, bug tracking services, knowledge bases, etc. These services cover most elements of the software development cycle.
- search and distribution services, e. g. maintaining a software archive which provides OSS products for download.
- value added services for OSS products, e. g. complementary services like consulting, documentation and systems integration.

In order to gain effects of scale, intermediaries cover a specific range of OSS products (*product portfolio*) and provide services via internet portals. Typically, some of these services are provided for a fee in order to gain profit. This approach has led to a vast multitude of business models for OSS (see [12], p. 50-55, and [8], pp. 5-12), like hardware- and software integration, technical support and publications, contract development, consulting and training. Fig. 2 provides a review of these interrelationships.

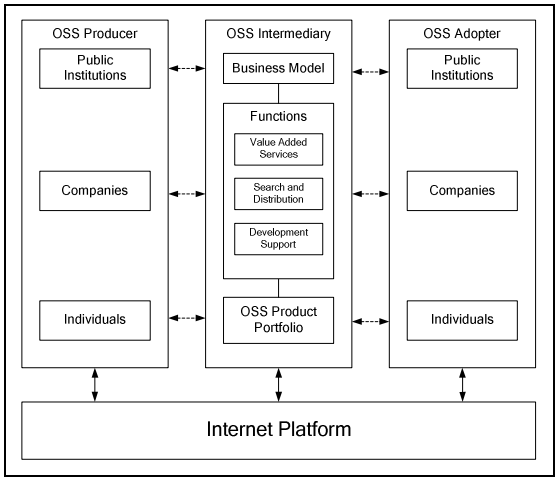

Fig. 2. OSS intermediaries

Several intermediaries emerged and some of them already disappeared from the OSS market, because they failed to fulfil their financial goals (see [15] and [1]). A major factor for this shortcoming lies in coordination problems and failures, which emerge from the OSS development model (see [14], p. 77). In order to handle these coordination deficiencies, it is necessary to design and deploy controlling instruments which are able to support and enhance market-related decisions of OSS intermediaries. These instruments should generate relevant information about market-related processes to ensure rationality of management [22].

As depicted in fig. 2, all actors of the OSS market are connected via internet. Since many market-related processes are mediated by portals, it is possible to technically observe interaction behaviour of all involved parties. Especially in the case of a highly centralised portal – for instance SourceForge.net – collection of monitoring data is rather effortless.

A promising means of data acquisition is server logging, which generates a log file containing all elementary interactions of the OSS community with the intermediary's internet portal. Since this is an unobtrusive data acquisition instrument, the behaviour of market actors can be recorded without subject or observer biases. In addition, data acquisition is cost-effective, because a vast multitude of web servers support logging mechanisms a priori. In order to harness this data source for controlling purposes, it is necessary to investigate its structure and to identify adequate concepts for analytical processing.

3. Web log mining

3.1 Server based log files

As the name suggests, the concept of Web Log Mining deals with the analysis of web server log files. These log files record all activities of a web server on the level of the hypertext transfer protocol (http), such that all interactions of site visitors with the server are continuously recorded by a series of entries. These entries document time-referenced events which correspond to visitor's online behaviour, e. g. requesting specific html pages from a web server, downloading software or filling out online forms [4].

Technically speaking, a server log file is a regular ASCII data file, whose structure is subject to different factors. On the one hand, there are different log file formats which determine file structure and content. On the other hand, the server operator is able to customise the log file format to meet his own preferences. In order to examine the informational potential of log files for management support of open source intermediaries, we will focus on the *combined log file format* which is used by the apache http server. This is an open source product which has reached a high degree of diffusion – in particular within the open source community – and thus represents a focal platform. Figure 3 describes the fields of the combined log file format.

	Field name	**Description**
1	host	IP/DNS address of the http client that made the request.
2	rfc931	Client identification according to rfc 931.
3	authuser	Login name used by the client for authentication.
4	datetime	The date and time stamp of the http request, and the timezone of the server.
5	request	The http request contains the requested resource (e. g. “index.html”), the http method (e. g. “GET”) and the http protocol version (e. g. “1.1”).
6	status-code	This field indicates the success or failure of the http request.
7	bytes	Number of bytes transferred.
8	referrer	The URL the client visited before coming to the website.
9	user-agent	Web browser and operating system used by the client.

Fig. 3. Combined log file format

The description of the attributes reveals that log file data primarily focuses on technical issues and is weakly structured. Therefore, it is merely possible to analyze log files instantaneously to gain insight into visitor behaviour. Consequently, it is necessary to carry out different steps to prepare log file data. These steps are formalised within the web log mining process model, which is discussed in the following section.

3.2 Web log mining process model

In order to derive relevant information from web server log files, it is initially necessary to select, clean, enrich and format log data. After this, it is possible to apply mathematical or statistical methods in order to extract relevant patterns from log data. These patterns have to be evaluated by domain experts and might be deployed in order to support decisions at the operational level. All these steps together build the superordinate concept of *data mining* which is an integrated process of gaining information from data [2]. In the context of log file data, the concept of *web log mining* applies [3]. The structure of the web log mining process is shown in fig. 4.

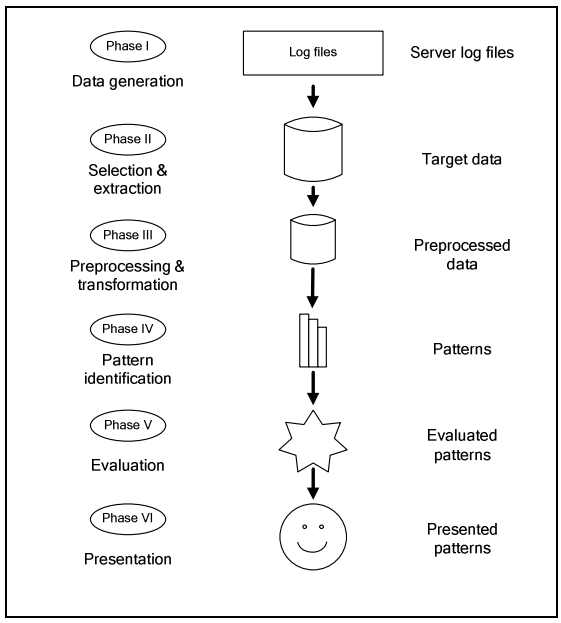

Fig. 4. Web log mining process [2]

The first step of web log mining deals with data generation, i. e. the log files are created and updated by the web server handling incoming http requests. For analytical purposes, it is necessary to select and extract relevant log file data. These activities are subject to situational objectives, such that data extraction is driven by temporal or other factual criteria. Since log file data are weakly structured, it is necessary to preprocess log files. For the analytical objectives of web log mining, the reconstruction of user paths through the site is essential. This task is also known as *sessionizing* (see [5] and [13]).

Subsequently, methods are applied to the log file data in order to discover potential relevant patterns. For instance, following techniques can be used (see [23], p. 16):

- descriptive statistical analysis to calculate interesting measures like number of visits or visit duration.
- association rule analysis to discover dependencies between objects (e. g. web pages or products).
- clustering to identify segments of users exhibiting similar behaviour.
- sequence analysis to identify significant browsing paths.

In order to avoid information overload, irrelevant patterns have to be erased from the result set (evaluation). Afterwards, the patterns are visualised, such that decision makers are able to interpret and deploy the results.

3.3 Web log mining in e-learning domain

This brief description illuminates the generic steps of the web log mining process, but does not deduce anything about its informational potential to support coordination tasks. In order to substantiate domain specific benefits, we evaluate the domain of e-learning. In this domain, two main categories of actors are to be differentiated, e-learning users (learners) and e-learning providers (e. g. universities, private training companies). By use of web log mining, it is possible to identify behavioural patterns of these actors, provided that log file data is available. Since many institutions, especially distance and presence universities, provide network centric e-learning platforms (e. g. learning management systems, LMS), massive log file data sets are available. Consequently, web log mining can be applied to get insight into student behaviour. For instance, following patterns can be identified:

- number of student visits and downloads of e-learning resources.
- associations between requested e-learning resources. These patterns can be used to con-

struct composite e-learning offers (product bundling).

- identification of student groups with similar learning behaviour. These groups can be used for market segmentation and provision of customized e-learning offers.
- identification of dominant e-learning sequences and learning paths. These sequential patterns serve to optimize the navigational structure of e-learning resources.

These patterns underline the usefulness of web log mining for the operators of learning management systems (e. g. single universities or faculties). Since the open source paradigm has lasting impact on the e-learning sector as a whole, it is necessary to evaluate the informational potential of web log mining for the providers of open source e-learning solutions. In the following, we present an innovative application study of web log mining which focuses an intermediary acting as provider of open source e-learning products and services.

4. Web log mining case study – the OSS intermediary CampusSource

4.1 The OSS intermediary CampusSource

Aim of the intermediary *CampusSource*, which was founded by the Ministry of Science and Research of the Federal State of North Rhine-Westphalia (Germany), is to set up cooperative processes for the development of software systems and modules as well as the creation and operation of an infrastructure for e-learning [9]. The efforts of single university projects are brought together and Open Source platforms as technical requirement of presence- and distance-universities are provided for use and further development to everyone interested [11]. Since most of the platforms are available in different languages, CampusSource provides services for a global e-learning community. The product portfolio of this intermediary currently consists of 15 e-learning products and is extended in consideration of a severe quality policy. Intermediary functions of CampusSource are:

- to support development processes, e. g. quality control and user feedback, and to supply OS related domain knowledge fostering system adoption and development in the field of e-learning,
- to provide e-learning products for download (distribution) and to manage licensing, in particular with regard to the legal peculiarities of OSS (e. g. license compliance to national intellectual property right regulations), and
- to broker value added services for e-learning products, e. g. complementary services like consulting, server hosting and systems integration.

All functions are provided by an internet portal accessible through the web site http://www.campussource.de/org/.

4.2 Outline of the study

In order to extract potential useful knowledge from the log files of CampusSource.org, the study followed an exploratory approach. This was necessary, since the informational relevance of the log files for decision making was quite unclear. So far, just registration data was analysed in order to assess diffusion of the product portfolio provided.

The data base of the study consists of log files stored in the apache combined log file format covering all online interactions from 09-01-2001 until 03-31-2003. We used standard software tools as analytical platform to carry out all steps of the web log mining process [19] and focused on descriptive statistical analysis to get a first insight into the available data. Of course, the results may lead to complex questions which could be answered by use of more sophisticated methods (e. g. inferential statistics).

For the reason of space, we merely present a subset of our findings which serves to expatiate the benefits of web log mining.

4.3 Results

4.3.1 Web site reach

In order to evaluate the reach of the CampusSource website, we analysed the total number of visits in a first step. A visit is defined as a temporal coherent sequence of interactions exerted on the website by a single user. This key figure shows the contacts of potential adopters with the CampusSource website, but does not evaluate the quality of the contact (e. g. whether the provided information does match the preferences of the visitors). Therefore, we used the visit duration as general indicator for the attractiveness of the

website. Fig. 5 depicts the results for the key figures *visits* and *duration* per quarter.

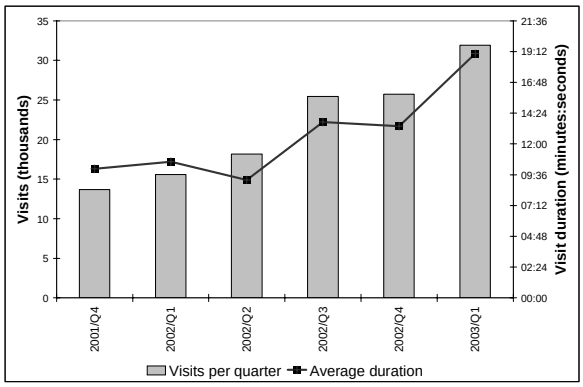

Fig. 5. Visits and duration per quarter

It is discernible that there is a significant increase in visits and duration over time. Roughly speaking, both figures have doubled from 2001 to 2003:

- In the last quarter of 2001, the number of visits amounts to 13,684, in the first quarter of 2003 this key figure amounts to 31,903 visits.
- In the last quarter of 2001, the average duration amounts to 10:04 minutes, in the first quarter of 2003 the average duration amounts to 19:02 minutes.

This result does confirm the increasing overall attractiveness of the CampusSource website. From a strategic point of view, this growth provides the basis for gaining network effects by rapid diffusion and should be measured continuously in order to signal negative trends. Nevertheless, this result just tells something about the quantity and duration of contacts ("reach"), but not how these contacts occurred. In order to provide information for optimizing search and distribution functions (see fig. 2), we tackled the subject of search behaviour.

4.3.2 Search behaviour

For the management of an OSS intermediary it is interesting to see how users establish contact to the website. This knowledge is essential in order to attract potential OSS adopters via internet channels. Consequently, we analysed the referrer field, which indicates the website the user visited *before* he browsed to the CampusSource site (see fig. 3). Fig. 6 shows the top 10 referrers. The shares of the top referrers relate to the total number of visits from quarter 1 in 2001 to quarter 1 in 2003 (130,531 visits).

No.	Top referrer	Share
1	http://www.google.de/	51,88 %
2	http://www.google.com/	19,46 %
3	http://openuss.sourceforge.net/	8,09 %
4	http://www.ilias.uni-koeln.de/	4,59 %
5	http://www.gmd.de/	3,08 %
6	http://www.fernuni-hagen.de/	2,76 %
7	http://www.google.ch/	2,73 %
8	http://www.google.at/	2,24 %
9	http://www-pi3.fernuni-hagen.de/	1,95 %
10	http://muckel3.fernuni-hagen.de/	1,66 %

Fig. 6. Top referrer

It is notable that more than 75 % of the website visits were established via the google search engine (no. 1, 2, 7 and 8). This shows that the majority of visitors make use of a keyword driven, general purpose search engine to locate the CampusSource website and therefore show an intentional search behaviour. On the other hand, a fraction of 8 % comes via the OSS intermediary SourceForge.net (no. 3). This is not a striking fact, since some of the products provided by CampusSource are hosted at SourceForge and reference the CampusSource website. These visits may correspond to users who have identified (technical) OSS product information at SourceForge (e. g. source code, bug fixes, etc.) and are looking for complementary services (e. g. installation and integration). Of course, this hypothesis has to be tested by more complex analytical methods. For instance, sequence analysis can identify significant browsing paths of visitors coming via SourceForge. Knowledge about SourceForge visitors spilling over to CampusSource is potentially useful to provide customised product offerings, e. g. value added services for specific e-learning products hosted at SourceForge.net.

It is also interesting to know which search terms were used by the visitors to locate the CampusSource website. To find this out, we extracted search terms from the referrer field. Most search engines store the search words entered by the user within the request field (see fig. 3), such that it is possible to gain insight into expressive user interests leading to the CampusSource web-

site. Fig. 7 lists top search terms extracted from the referrer field.

No.	Search term
1	e-learning
2	openuss
3	Virtuelles lernen („virtual learning")
4	ilias
5	webassign
6	campussource
7	opensource
8	wie funktioniert die börse („how the exchange works")
9	software exchange
10	open source

Fig. 7. Top search terms

Fig. 7 makes clear that most of the visitors use domain specific or product specific search terms. For instance, the terms *e-learning*, *virtual learning* and *open source* (no. 1, 3, 7 and 10) indicate interest in the fields of e-learning or open source. On the other hand, product names like *openuss*, *ilias*, *webassign* (no. 2, 4 and 5) – which all correspond to elements of CampusSource's product portfolio – trigger contact to the website.

This result is extremely useful in order to examine if the "right" users are attracted by the website, which seems to be the case for CampusSource. In addition, it also gives an indication for the existence of different user segments. For instance, there might be users focusing the domain of e-learning, while others are searching for background information about open source. This hypothesis may be tested by more complex analytical instruments, for example by use of clustering techniques.

As a whole, the results of the search behaviour analysis provide valuable knowledge for CampusSource. This knowledge can be utilised in order to personalise information streams dependent on specific visitor interests. This personalization may use basic operations like customizing, filtering, annotating or aggregating information (see [17], p. 98) about OSS products or value added services. Finally, this contributes to a higher degree of coordination between user preferences and offerings based on search behaviour.

4.3.2 Product portfolio

Currently, CampusSource covers a portfolio of 15 e-learning software products, which are provided for download on the website. This product portfolio is extended by new OSS products on occasion. To support product portfolio management, it is necessary to track the diffusion of the provided products. In order to measure diffusion on the product level, we analysed the download rates of the OSS products, usually supplied as compressed archives (e. g. zip or tar format). Figure 8 depicts the number of downloads for some of the core products of CampusSource and the total downloads per quarter.

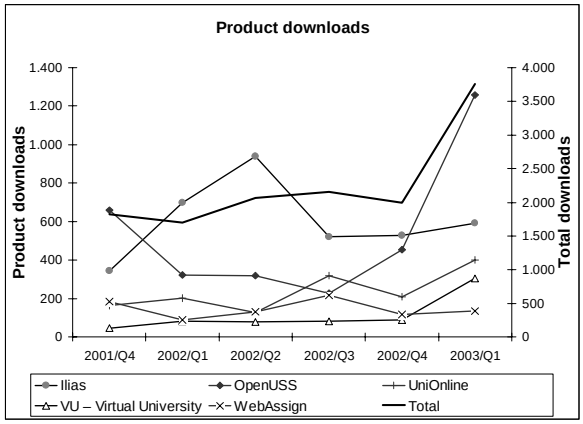

Fig. 8. Product downloads per quarter

On the individual product level, the download measure is an overall indicator for product diffusion and is therefore relevant for the intermediary's product policy. For instance, a steep increase in downloads for an individual product potentially indicates a growing demand for complementary products and services. Therefore, download rates are basically interesting for product related decisions. Of course, this measure has to be interpreted carefully because the vast majority of downloads does not imply successful product adoption (i. e. download *and* productive use of an OSS e-learning product). Therefore, these download figures should be regarded as upper bound of product diffusion. In order to measure the diffusion rate, additional data acquisition instruments are necessary (e. g. obligatory registration forms).

It is interesting to see that the individual download rates are subject to significant fluctua-

tions over time. As we could notice, these fluctuations heavily depend on individual product policies, e. g. the availability of a new software release. As a consequence, timing of the provision of additional value added services is potentially crucial.

5. Conclusion

This paper proposed a controlling instrument for OSS intermediaries based on behavioural data (log files). On the basis of the theoretical foundations of the economic application field (OSS intermediaries) and the analytical framework (web log mining), we finally substantiated the usefulness of this instrument in the context of an OSS intermediary for the domain of e-learning. These empirical findings showed that web log mining establishes transparency concerning online behaviour of OSS adopters and therefore creates management information necessary for coordination of OSS related decisions. These decisions concern the OSS product portfolio, the provision of suitable search functions and complementary value added services, which are predominantly relevant for intermediaries to yield a profit.

Further research is necessary in order to enhance the utility of this controlling instrument for intermediaries. On the one hand, this implies the application of more complex analytical methods, e. g. clustering and sequence analysis. On the other hand, deep integration into management systems is required, so that technical and organisational aspects have to be addressed in equal measure. For instance, the web log mining process has to be adapted in order to meet the information demand of the management. Consequently, integration into reporting in consideration of situational information quality criteria has to be examined carefully. It is assumed that this is essential to enhance the coordinative competency of OSS intermediaries to ensure achievement of financial and non-financial objectives in the long run.

6. Acknowledgements

The authors would like to thank Mr. Tobias Ortmann for his profound technical engagement in the work for the exploratory study on CampusSource [19].

7. References

[1] Bensberg, F., Dewanto, B. L., Wandel im Softwaremarkt durch Open Source, in: Software Development, Vol. 1, No. 3, pp. 51-55.

[2] Bensberg, F., Web Log Mining als Instrument der Marketingforschung, Wiesbaden (Germany), 2001.

[3] Bensberg, F., Weiß, T., Web Log Mining als Marktforschungsinstrument für das World Wide Web, in: Wirtschaftsinformatik, Vol. 41, No. 5, 1999, pp. 426-432.

[4] Büchner, A. G., Mulvenna, M. D., Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining, in: SIGMOD Record, Vol. 27, No. 4, 1998, pp. 54-61.

[5] Cooley, R., Mobasher, B., Srivastava, J., Data Preparation for Mining World Wide Web Browsing Patterns, in: Knowledge and Information Systems, Vol. 1, No. 1, 1999, pp. 5-31.

[6] Feller, J., Fitzgerald, B., A Framework Analysis of the Open Source Software Development Paradigm, in: Proceedings of the International Conference on Information Systems (ICIS), December 10th-13th, Brisbane (Australia), 2000, pp. 58-69.

[7] Friedewald, M., Blind, K., Edler, J., Die Innovationstätigkeit der deutschen Softwareindustrie, in: Wirtschaftsinformatik, Vol. 44, No. 2, 2002, pp. 151-161.

[8] Hohensohn, H., Hang, J., Product- and Service Related Business Models for Open Source Software, 2003. http://mysite.fh-coburg.de/~wielandt/OSSIE03/ossie03-HohensohnHang.pdf [01/28/2004].

[9] Hoyer, H., CampusSource – an Open Source Initiative of the MWF NRW, http://www.-campussource.de/org/about/ [01/27/2004].

[10] Janssen, M., Sol, H. G., Evaluating the Role of Intermediaries in the Electronic Value Chain, in: Internet Research: Electronic Networking Applications and Policy, Vol. 10, No. 5, 2000, pp. 406-417.

[11] Kaderali, F., Ehlert, O., Experience with Open Source for e-learning, EADTU Annual Conference Madrid, "E-Bologna" – Progressing the European Learning Space, Madrid (Spain), http://ks.fernuni-hagen.de/forschung/veroeff/pubs/oe.eadtu2003/eBologna-03-KaderaliEhlert.pdf [01/28/2004].

[12] Karels, M. J., Commercializing Open Source Software, in: Queue, July-August 2003, pp. 46-55.

[13] Kohavi, R., Mining E-Commerce Data: The Good, the Bad, and the Ugly, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001. http://robotics.stanford.edu/~ronnyk/goodBadUglyKDDItrack.pdf [01/28/2004].

[14] Kooths, S., Langenfurth, M., Kalwey, N., Open-Source Software – An Economic Assessment, MICE Economic Research Studies, Vol. 4, Münster 2003.

[15] Kulikauskas, A., Ellison-Bey, D., An Economy for Giving Everything Away. http://www.ms.lt/en/workingopenly/givingaway.html [12/27/2003].

[16] J. F. Santiago Luque, A. Alabau, European Union and the Open Source/Free Software, 2003.

[17] Maglio, P., Barret, R., Intermediaries Personalize Information Streams, in: Communications of the ACM, Vol. 43, No. 8, 2000, pp. 96-101.

[18] Malone, T. W., Yates, J., Benjamin, R. I., Electronic Markets and Electronic Hierarchies, in: Communications of the ACM, Vol. 42, No. 3, pp. 81-91.

[19] Ortmann, T., Analyse des Nachfragerverhaltens von Open Source Software – dargestellt am Beispiel des OS-Mediatoren Campus-Source, Term Paper, University of Muenster, 2003.

[20] Open Source Initiative, The Open Source Definition. http://www.opensource.org/docs/definition_plain.html [01/18/04].

[21] Quah, D., Digital Goods and the New Economy, in: New Economy Handbook, D. C. Jones (Ed.), New York et al. 2003, pp. 289-321.

[22] Schäffer, U., Weber, J., Prenzler, C., Characterising and Developing Controller Tasks – A German Perspective, 2001. http://www.whu.edu/control/Forschungspapiere-/FPCCM03.pdf [01/28/2004].

[23] Srivastava, J., Cooley, R., Deshpande, M., Tan, P.-N., Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, in: SIGKDD Explorations, Vol. 1, No. 2, 2000, pp. 12-23.

[24] Weitzel, T., Beimborn, D., König, W., Coordination in Networks: An Economic Equilibrium Analysis, in: Information Systems and e-Business Management, Vol. 1, No. 2, 2003, pp. 191-211.