A Web-based Mechanism to Avoid Mispricing Products on E-Commerce Platform

Tai-Chun Wang[1]
[1]Advanced Research Institute, Institute for Information Industry, Taipei, Taiwan
tcwang@iii.org.tw

***Abstract*- Online mispriced product incidence has been increased dramatically in recent years. Unlike brick-and-mortar retailers, pricing error brings more financial or custom relation loss to online shopkeepers. Hence, to avoid significant losses caused by mispricing products, it could be a benefit if a vendor can prevent this human carelessness during stocking online. In this article, we present a misprice preventing mechanism which based on the concept of web price mining. Via using business process modelling, our method has the flexibility to replace any algorithm components as well as integrated with any reference sites easily. Experimental results indicate that our mechanism can prevent mispricing product in most cases.**

I. INTRODUCTION

The mispricing product incidents on the online shop have been increasing dramatically in Taiwan in recent years. Unlike brick-and-mortar retailers, pricing error brings more financial or custom relation loss to online shopkeepers. It is because thousands of orders can be placed before online retailers detect the problem [3]. Since most companies who encoder this issue refused to fulfill those orders, damaged has been made for not only their reputations, but also the loyalty of their customers. Hence, it could be a benefit if a vendor can prevent this human carelessness during stocking online.

In order to acquire the reference price of a product, researchers have introduced several algorithms which based on web mining to handle this issue [7, 8]. Web mining technology involves framework mining, context mining and log mining [4]; and is used widely for commercial purposes, e.g. user behavior [1], news discovery [6], analyze social network connections, terrorist threat detection [2] and usage mining [5], etc. A web mining algorithm builds a mining target model and use program to extract information from markup language or through an application programming interface (API) directly [7]. In addition, Web price mining has been used as a business strategy for price comparison [7] and price forecast recently [8].

In this study, we introduce a mechanism for preventing mispricing products online. This mechanism is based on the web price mining concept and can be implemented as an assistant function/plugin for any E-Commerce platforms. In order to make our mechanism flexible to adopt any E-Commerce platform or replace price mining algorithm easily, we implemented this service via business process modeling notation.

II. PRELIMINARY

The price query process presented in this study involves two computational technologies: “Business process model and notation” and “application programming interface.”

A. Business Process Model and Notation (BPMN)

Reusable process is one of the core features of the smart system service. BPMN is a standard format for graphical representation of business process model. It defines the elements, like flow objects and connecting objects, etc., to

DOI: 10.978.819252/12227

form a process. Several commercial products [9-11] can be used to design and implement business process into the information technology level.

B. Application Programming Interface (API)

In computer programming, an API is used to specify the interaction method between different software components. When an API is implemented as a web service, it accepts a set of Hypertext Transfer Protocol (HTTP) request messages and returns any types of data object. The most common return data object format is Representational State Transfer (RESTful) or JavaScript Object Notation (JSON).

III. METHOD

In this section, we first present how we implement our mechanism followed by the structure/steps of our mechanism.

C. System Implement

In this study, we choose Bonita to implement the API. In addition, we also design a Google Chrome web browser plugin to examine the mechanism on an E-Commerce platform. A central server is established for our API to provide the price referring service. A database is used to store all the pricing information, includes product name, price, URL and latest query time of products which has been queried by someone before. We also select Yahoo! (tw.mall.yahoo.com), PChome (hwww.pcstore.com.tw) and Postmall (postmall.post.gov.tw/postmall/) as the price reference E-Commerce platforms (which have been implemented as a node in the process) because each of these three platforms covers the most of common products.

D. Price inquiring process

The price inquiring process includes 6 steps (see Fig. 1):

1) Price inquiry: User sends a price query request via the API. The input data could be the product title only or includes the product description.

2) Product name extraction: When the server receives the request, it will analyze the input data to figure out which product's price the user is seeking for.

3) Finding exist information: Based on the result obtained from step 2, our process will check if this product has been queried before through seeking it in our database.

Case 1. If the product in not in our database, the server will query its price from the selected E-Commerce platform directly.

Case 2. If the product is stored in our database, our process will start the "update price information" sub-process (see subsection F).

4) Store information: The process will save the price information and the related product URL which obtained from those preselected E-Commerce platforms into the database.

5) Return the price information.

In step 1, after employing our plugin, users first fill in the product title and the product description fields. The plugin will detect user activities and trigger the price querying API automatically. It is noteworthy that, our process

DOI: 10.978.819252/12227

does not restrict the user to use a precise product name while calling the API. For example, the API accepts a title term like "A must have Paul Smith watch once in a lifetime", rather than just use "Paul Smith watch (model xx-yyy)". We deploy the CKIP Chinese Word Segmentation System introduced by Academia Sinica (Taiwan) [12] to extract the possible product name that the user is queried for. With the advantage of using process engine, this algorithm is replaceable by other syntax or pattern analysis algorithms.

The core step, step 3, is used for querying the product price from those preselected E-Commerce platforms. We used APIs that provided by those platforms to handle the price query task. It is noteworthy that, we obtain multiple feedbacks from individual platforms in most of the queries. In that case, we used the algorithm that we used in step 2 again to compare these feedbacks with the product name we are asking for in order to find the perfect matches. The product may or may not be queried before. Thus, the process would need to deal with those two cases in different ways. We will describe how we are solving these two cases later in section E.

All query results will be stored in our database (step 4) in order to reduce the query time for the same product. Each record includes the product name, the web URL of this product, the price as well as the latest query time.

E. Update price information sub-process

A record would be stored in our database if a product which a user is querying has been queried by someone else before. In order to keep the record up-to-date, process will refresh this record via re-obtain the price information through five steps as follows:

1) *Use the URL information from the database to obtain the new price again.*
2) *Check the product name shown on the current web page that directed by the URL matches the product name we are querying for.*
3) *If the product name matched, update the price information.*
4) *If the web page that the URL pointed to does not exist anymore, re-query the platform again to get the new product page as well as its URL. After that, go to step 3.*
5) *Return the price information.*

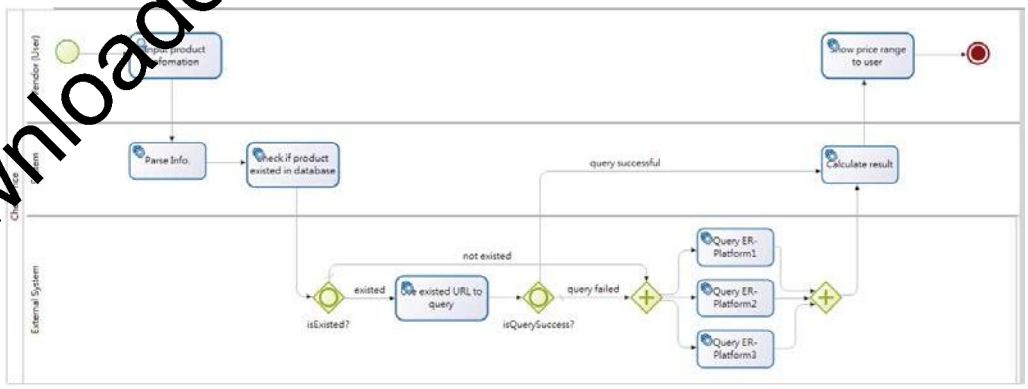

Figure 1. The price querying process implemented by BPM.

DOI: 10.978.819252/12227

F. User interface

As we descripted before, we implemented the price querying process as a web browser plugin. In this section, we will illustrate how this process works as a mechanism for the prevention of mispricing. It is noteworthy that, although we demonstrate this mechanism via web browser plugin (Greasemonkey), this mechanism can also be embedded as part of the E-Commerce platform's function through calling our API directly from the web page.

The plugin keeps monitoring the product title and price fields on the product stocking form. The price querying API will be triggered once when a vendor fills in the product title. After the API gets the reference price from the other EC website, the plugin will compare the reference price with the amount the vendor fill in. If the vendor inserts a price which is lower than the price range given by our API, warning information will be shown on the web to inform the vendor to modify the price (see Fig. 2). We also suggest the developer to lock the form submit button for the vendor.

IV. CONCLUSION

Online mispriced product incidence brings the loss of business reputation of a company. In any crisis management strategy will come with certain drawback or side effect; especially it is against the law to cancel these orders in some country. Hence, to avoid significant losses caused by mispricing products it could be a benefit if a vendor can prevent this human carelessness during stocking online. In this article we present a misprice preventing mechanism to deal with this issue. Via using BPMN to implement this mechanism, our method has the flexibility to replace the syntax parsing algorithm as well as add or remove reference sites easily from the process. It is noteworthy that, people who want to build up their own application based on our mechanism can/should select those tools and/or programming languages they are familiar with. Experimental results (data not shown) indicate that our mechanism can prevent mispricing product in most cases. Thus, our mechanism should be beneficial to e-shop holders. In the future, we will focus on dynamic selection of referring EC website so that developers will not need to modify it in the BPMN process.

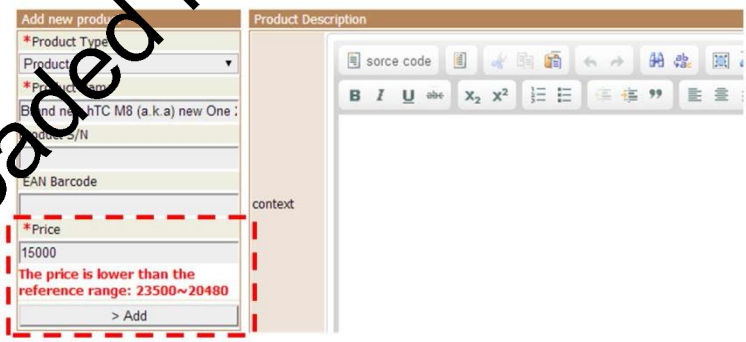

Figure 2. The illustration of suspicious price warning.

ACKNOWLEDGMENT

This study is conducted under the "Smart System Service Technology and Business Transformation Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China. I personally thank Mr. Max Chen's great help for literature surveying.

DOI: 10.978.819252/12227

REFERENCES

[1] S. Antony, P. Wu, D. Agrawal and A. El Abbadi, "Aggregate skyline: Analysis for online users," in *Applications and the Internet*, 2009. SAINT'09. Ninth Annual International Symposium on, 2009.

[2] H. Chen, "IEDs in the dark web: Lexicon expansion and genre classification," in *Intelligence and Security Informatics*, 2009. ISI'09. IEEE International Conference on, 2009.

[3] B. Groebner, "Oops! The Legal Consequences of and Solutions to Online Pricing Errors," *Shidler JL Com. & Tech.*, vol. 1, p. 1, 2004.

[4] C. Qi and H. Ming, "XML-based data mining design and implementation," in *2010 International Conference On Computer Design and Applications*, 2010.

[5] S. Salin and P. Senkul, "Using semantic information for web usage mining based recommendation," in *Computer and Information Sciences*, 2009. ISCIS 2009. 24th International Symposium on, 2009.

[6] C.-Z. Xu and T. I. Ibrahim, "A keyword-based semantic prefetching approach in Internet news services," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 5, pp. 601-611, 2004.

[7] Q. Zhu, H. Zhou, Y. Yan, J. Qian and P. Zhou, "Commodities Price Dynamic Trend Analysis Based on Web Mining," *2011 Third International Conference on Multimedia Information Networking and Security*, 2011.

[8] Q. Zhu, S. Cao, J. Ding and Z. Han, "Research on the Price Forecast without Complete Data based on Web Mining," in *Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2011 Tenth International Symposium on*, 2011.

[9] Bonitasoft, http://www.bonitasoft.com/. (Accessed on March, 2014).

[10] WebSphere Lombardi Edition, http://www-01.ibm.com/software/integration/lombardi-edition/. (Accessed on March, 2014)

[11] RunMyProcess, https://www.runmyprocess.com/en/. (Accessed on March, 2014).

[12] Academia Sinica (Taiwan), "CKIP Chinese Word Segmentation System," http://ckipsvr.iis.sinica.edu.tw/ (Accessed on January, 2014).

DOI: 10.978.819252/12227