Technische Universität Dresden
Medienzentrum

Prof. Dr. Thomas Köhler
Jun.-Prof. Dr. Nina Kahnwald
(Hrsg.)

GeNeMe '13

Gemeinschaften in Neuen Medien

an der
Technischen Universität Dresden
mit Unterstützung der

BPS Bildungsportal Sachsen GmbH
Campus M21
Communardo Software GmbH
Dresden International University
eScience – Forschungsnetzwerk Sachsen
Gesellschaft der Freunde und Förderer der TU Dresden e.V.
Gesellschaft für Informatik e.V.
Gesellschaft für Medien in der Wissenschaft e.V.
IBM Deutschland
itsax – pludoni GmbH
Kontext E GmbH
Learnical GbR
Medienzentrum, TU Dresden
ObjectFab GmbH
Transinsight GmbH
T-Systems Multimedia Solutions GmbH
Universität Siegen

am 07. und 08. Oktober 2013 in Dresden

www.geneme.de
info@geneme.de

B.4 SENSE: Combining Mashup and HSM technology by semantic means to improve usability and performance

Stefan Haun[1], Robert Krüger[2], Peter Wehner[2]
[1]Otto-von-Guericke-University, Data and Knowledge Engineering Group
[2]Fink & Partner Media Services GmbH

1 Introduction

The amount of data stored and consumed on a daily basis as well as the complexity of the data structure have grown rapidly in past years [1]. Especially business companies try to reduce the rising expenses from storage infrastructure as well as from re-implementation of user interfaces to adapt to evolving tasks.

Mash-Up Frameworks and systems for Hierarchical Storage Management (HSM) are two technologies focusing right on this need. Additionally, the domain in which data is used yields information about this data at a certain point in time, e.g. such as relevance or importance to the user base. The SENSE project tries to combine those aspects to enhance the overall system functionality by adding semantic technology and by that allowing for a self-organized storage (done by the system) as well as for advanced self-structured user interfaces (done by the user).

This paper describes a setup featuring a Widget/Dashboard-Framework called NewsDesk and the runtime-independent communication frameworks GLUE and MOCCA. We show how these parts, combined into the SENSE framework, can lead to increased performance regarding aspects like load time of data objects and overall storage load of a multi-tier architecture.

First, a short summary of related work is given followed by introducing the reader to the SENSE framework containing all the aforementioned components. Afterwards, the power of the synergy of all those components is illustrated by presenting a scenario utilizing domain knowledge from a collaborative image selection process in the media press domain within a platform-wise heterogeneous system environment. Eventually the paper is concluded with the presentation of the achieved results, as well as a short outlook on future development and research.

2 Related Work

To avoid re-implementing user interfaces over and over again the separation of UI building-blocks into so called widgets has been proposed. Sire et al. describe a dashboard as a place in the user interface that offers limited space to compose a set of widgets [2]. This concept already has been successfully employed by industry scale applications like Inter:gator[1] or Oracle-Metalink[2] and on top is a feasible way to achieve personalized user interfaces to support the user whilst performing workflows to his taste.

To increase the acceptance of user-constructed user interfaces, enhancing the ease of runtime composition [3] and the description of capabilities of different mash-ups to automatically generate an instance of such [4] has been and still is the matter of scientific research. Furthermore the enhancement of Mash-up technology especially by adding a semantic annotation to mash-up components has been addressed [13]

Employing lots of different runtime environments to conclude a task, poses the need for a runtime independent communication paradigm. A mash-up of independent data sources, algorithm providers and a graph-interaction user interface has been developed during the BISON project. During this project GLUE/MOCCA was used to realize such a runtime independent communication and by that proved that the communication with GLUE/MOCCA is feasible in a mash-up application scenario [5].

3 The SENSE Framework

Allowing a system to take semantic description into consideration while self-organizing its underlying storage structure as well as providing additional information back to the user interface demands a framework. The SENSE project focuses on realizing such a framework by providing a software stack that is comprised of several software components which are, on the one hand, loosely coupled, on the other hand strongly tied by knowledge of data and interaction semantics throughout the whole system. To achieve a complete integration, all components of a system for interaction on big data stores are included: from the Hierarchical Storage Management (HSM) on the bottom to the user interfaces on top of the software stack. In the following sections we present the components most relevant for showing how improved performance and usability can be achieved by the strong semantic integration implemented in the SENSE framework.

[1] http://www.intergator.de/produkt/enterprise-search/dashboard

[2] https://supporthtml.oracle.com

3.1 SENSE Overview

SENSE, an acronym for „Intelligent Storage and Exploration of large Document Sets“, is an ongoing research project driven by three industry partners and two academic institutions. The main concept focuses on a continuous flow of semantically enriched data between different tiers of a loosely coupled software stack. Especially noteworthy is the fact that not only real world, but also technical domain annotations to actual data as well as domain specific concepts are considered.

For example, image-preview files are technical entities from the multimedia application domain. The semantic knowledge of their existence can be employed in the hierarchical storage tier to keep the preview file within the fastest layer of the HSM for immediate delivery, but storing the noticeable larger high-resolution images in a slower and therefore cheaper storage tier. In return the hierarchical storage management system may inform a presentation-tier application that the time of delivery (another technical property) for some file will be exceptionally long because the targeted file system media has to be remounted.

The SENSE consortium shares the opinion that introducing additional technical and domain specific semantic information to modern software architecture will lead to improved scalability and performance of large document collections over time.

To accomplish this goal SENSE features a semantic repository at the heart of its concept to serve the different interconnected system tiers through dedicated application and technical domain ontologies as shown by Figure 1: SENSE Framework architecture.

For easy and flexible user interface development a Widget-Dashboard Framework has been included. Advanced and cost efficient storage structures can be realized with the Hierarchical Storage Management System contained by the SENSE-Framework. Those basic components demand additional logic to realize the aspired continuous integration. Such as the HSM Observer, which takes care of keeping file location and domain knowledge in sync and by that enabling the HSM-Extension to compute suggestions for the HSM which files to keep in the fastest layers and which files to purge to lower storage tiers.

A central component, the Resource Manager, is introduced for import and export purposes to ensure no synchronization problems occur while entities are pushed into or removed from the SENSE-Framework. Finally all necessary functionality useful to application developers is wrapped in the so called SENSE-API allowing for easy integration of the SENSE-Framework within different use cases.This paper mainly focuses on aspects related to the UI-Framework and the HSM.

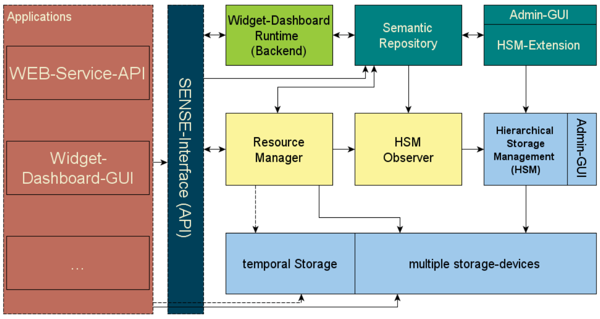
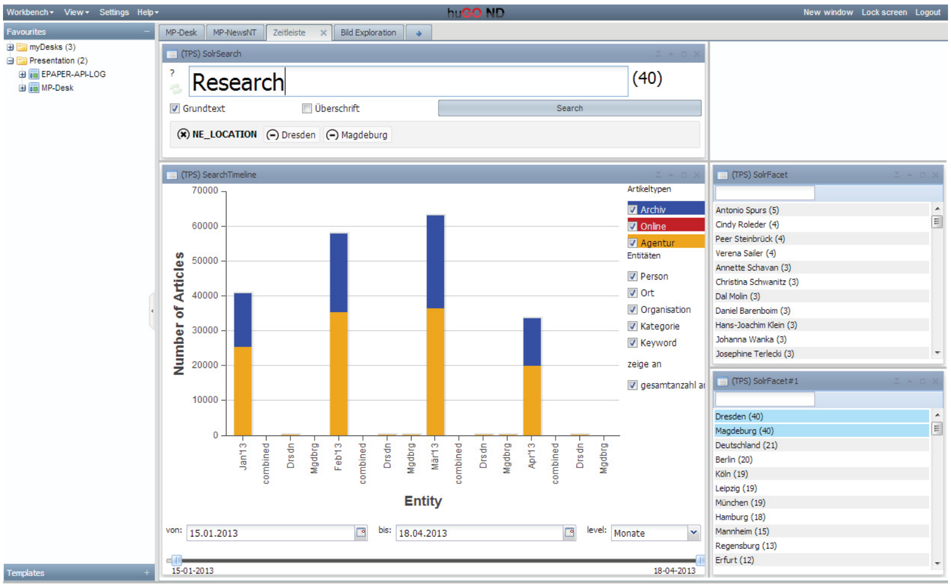


Figure 1: SENSE Framework architecture

3.2 NewsDesk: a Widget/Dashboard Framework for Mash-ups

NewsDesk is the aforementioned Widget/Dashboard-Framework which has been included in the SENSE-Framework. It is based on the web technology stack (HTML & JavaScript) and focuses on the approach of empowering end-users to compose their own user interfaces even at runtime. By that, end users can adapt the UI to fit the needs for the current tasks at hand.

Widgets are the basic building blocks within the NewsDesk framework which users can employ to construct their user interfaces on a so called dashboard. Widgets and dashboards are the core concepts of the UI-Framework and are enriched with means for easy reusability of once composed and configured dashboards and comfortable layouting [6]. An example of a dashboard containing four widgets is depicted by Figure 2: NewsDesk Exploration dashboard example.



Research (40)

☑ Grundtext ☐ Überschrift

NE_LOCATION: Dresden, Magdeburg

☑ Archiv
☑ Online
☑ Agentur

☑ Person
☑ Ort
☑ Organisation
☑ Kategorie
☑ Keyword

☑ gesamtanzahl a

von: 15.01.2013 bis: 18.04.2013 level: Monate

15-01-2013 18-04-2013

Antonio Spurs (5)
Cindy Roleder (4)
Peer Steinbrück (4)
Verena Sailer (4)
Annette Schavan (3)
Christina Schwanitz (3)
Dal Molin (3)
Daniel Barenboim (3)
Hans-Joachim Klein (3)
Johanna Wanka (3)
Josephine Terlecki (3)

Dresden (40)
Magdeburg (40)
Deutschland (21)
Berlin (20)
Köln (19)
Leipzig (19)
München (19)
Hamburg (18)
Mannheim (15)
Regensburg (13)
Erfurt (12)

Figure 2: NewsDesk Exploration dashboard example

Interaction between widgets in the form of communication is indispensable to realize business scale applications. NewsDesk bases the inter-widget-communication on a loosely coupled publish/subscribe approach [7]. NewsDesk-Widgets do not assume anything about other widgets they will be composed with on a dashboard. Hence widgets can be easily added or removed from a composition even at runtime. Communication-wise compatibility is automatically computed from a set of communication descriptions given by the widget developer.

The widget concept has been employed as provided by the NewsDesk from the beginning. Additionally to its original functionality the employment of the semantic repository provided by the SENSE Framework has been used to improve the automatic communication computation. Describing domain dependent typical user interface compositions has been used to generate composition proposals as well as an advanced wiring of communication wise compatible widgets [14]. Furthermore the knowledge of data types used by user interface components can be leveraged to deduce the most important data types and the once which are not employable or displayable at all while calculating recommendations for purging data to slower storage tiers by the HSM-Extension. This is one example of how knowledge about the user interfaces of a certain domain can be used to derive important information for the storage management components.

3.3 Communication

As the components of the SENSE framework are loosely coupled, may be distributed over several machines and, especially in the UI, require communication schemes more capable than the traditional request-response paradigm, we choose a novel concept for communication between the components. The lower-tier components, like the HSM and the HSM extensions as well as the semantic storage use the well-known XML-RPC for a request-response based communication. The higher-tier components, including the user interface, have more complex communication schemes like sending intermediate results, i.e. more than one responses to a request or additional meta-data, e.g. progress information that allows the UI to display a progress bar and an estimated time towards task completion. For those kinds of applications SENSE features GLUE and MOCCA to enable runtime independent communication.

GLUE - Generic Layer for Unified Exchange

GLUE is a Java library (with existing ports to different languages e.g. Javascript) that simplifies asynchronous communication between heterogeneous software components. It supports various exchangeable transport protocols such that data can be easily transmitted in every situation: within a single Java virtual machine (in-memory); over the wire (IP socket); or even using a chat protocol (XMPP). The chosen way of transmitting is based on the capabilities of the runtime the application employing GLUE is running on. GLUE provides a communication channel which is agnostic of the actual transport method and thus allows a flexible wiring of components in the SENSE framework.

MOCCA – Message-Oriented Command and Context Architecture

MOCCA is the Message-Oriented Command and Context Architecture, providing a GLUE-based middle-ware that allows sending commands to a peer which are executed by state-less handlers in a specific context. This context can be used to store and access data and will be provided with every call of those handlers and can be used, for example, to do effortless state modeling. In contrast to the request-response scheme the message flow is not fixed by the framework. This allows implementing additional communication paradigms. The whole system can be seen as an automaton with Messages that trigger state transitions in the local Contexts.

Within our scenario GLUE and MOCCA are used for communication between NewsDesk Widgets as well as between Widgets and external applications. This utilizes GLUES prominent feature to easily contact foreign platform services such as an android application or presentation software outside of the web scope. The feature index and the graph interaction components (both are described below) can also be seen as such. Finally there is a MOCCA agent for the SENSE API to bridge between MOCCA and XML-RPC communication.

4 Optimization through semantics

The following example from the press media domain illustrates the advantages arising from the described technologies combined in the SENSE framework. The scenario considers an image management system that employs the SENSE framework to provide access to a large amount, up to millions, of images.

Considering editors want to explore the dataset by means of similar images to find good candidates for a print publication and for similar tasks and eventually take their findings to an editorial meeting where they'll collaboratively select the images to be published in the next newspaper issue (Figure 3: Collaborative Image Selection).

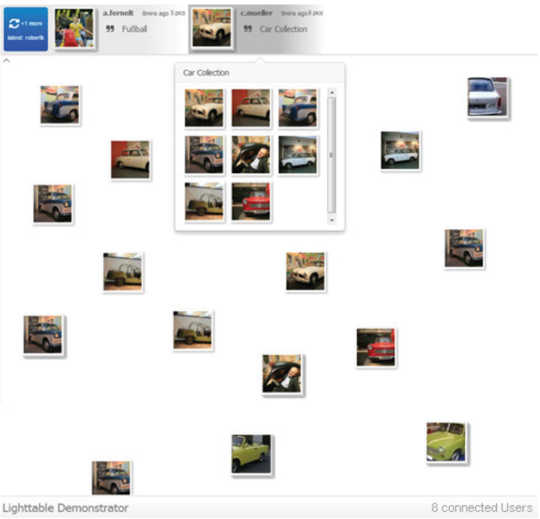
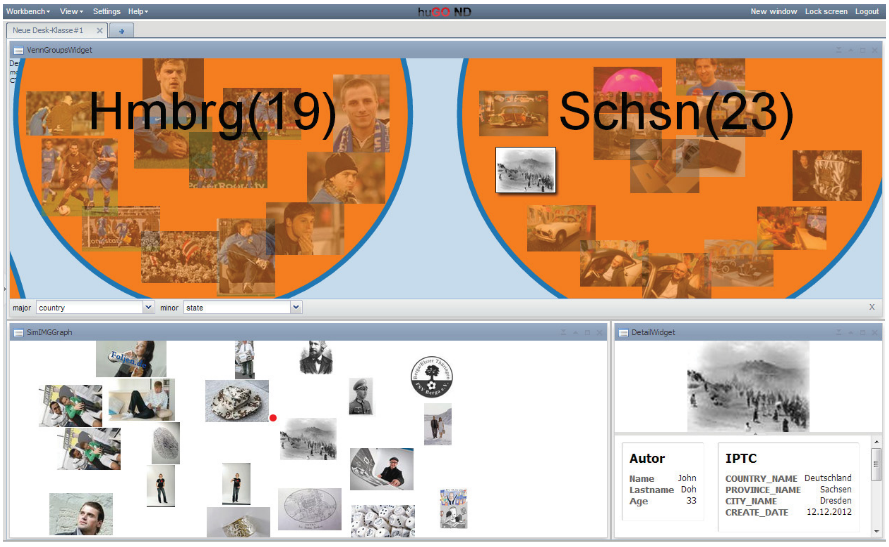


Figure 3: Collaborative Image Selection

Exploration is a search paradigm that, in contrast to an ad-hoc search like the well-known Google search, uses an iterative process that allows the user to define and refine an information need while discovering the search space until a sufficiently good result is found [8]. The scenario starts with a pivot image that has been acquired prior to the exploration by using available widgets. By employing NewsDesks widget/dashboard approach the user is free to choose how to obtain this pivot element. For example a user may favor a keyword search in image descriptions, a simple image list or widgets displaying groups of images as a Venn-diagram for a more explorative nature and so forth.

The pivot image is displayed on a canvas and can be expanded by the user. On expansion, an index-lookup for the top N similar images is performed, where N is a parameter in the exploration widget. The retrieved images are added to the canvas and all visible images are re-positioned, taking similarities and already displayed elements into account. The scenario is depicted by Figure 4: Image Exploration Scenario, Venn-Group Widget (top), Similar Image-Graph Widget (lower left), Image-Detail Widget (lower right).



major: country
minor: state

Autor

Name	John
Lastname	Doh
Age	33

IPTC

COUNTRY_NAME	Deutschland
PROVINCE_NAME	Sachsen
CITY_NAME	Dresden
CREATE_DATE	12.12.2012

Figure 4: Image Exploration Scenario, Venn-Group Widget (top), Similar Image-Graph Widget (lower left), Image-Detail Widget (lower right)

To enable this image exploration scenario, several components are incorporated: A feature index stores vectors in the feature space by which similarities are calculated. This allows pre-calculation, a necessary step to make the similarity search sufficiently fast to be used in a user interface. A metric that fits the feature space is used to calculate similarities when the query is evaluated. The feature index is quite independent from the selected feature, which may be a pure image feature like the Color Layout Descriptor [9] or a semantic feature like annotations about the image content or the image description.

The graph interaction component decouples the layout calculation, which decides the display positions of each image in the canvas, and the presentation of the visualization in the NewsDesk widget. Graph layouting is a vast topic with dozens of methods addressing specific visualization needs. However, most graph visualization methods are designed for static graphs. During an exploration the graph will change, i.e. nodes and edges are added or removed, and many methods show a chaotic behaviour regarding the difference in the graph and the difference in a layout. Humans, however, are well-suited to remember locations [10] and therefore an interaction-based graph layout must take care to keep the layout as stable as possible considering the changes in the graph. The layouting method uses a combination of multi-dimensional scaling (MDS) and node-overlap removal based on triangulation [11, 12]. The implementation works incrementally towards an optimal graph with respect to node positions and graph layouts already shown to the user. However, the calculation is very complex and the current implementation in Java is performance wise not suited for the browser.

Therefore the layouting has been split up: A java backend component stores the current graph in a MOCCA context and takes care of layout calculation. Only the resulting node positions are sent to the user interface component, which renders the graph on the screen. This frees the UI from the burden of delivering the current state all over again to the graph calculating process because during the exploration process the graph is restored from the MOCCA context on the java side and re-used for further calculation steps.

In an integrated system, like the SENSE framework, knowledge about the overall process can be leveraged to optimize the performance: When the list of similar images is returned by the Feature Index to the Graph Interaction component, it is most likely that this image, or a representation like a preview image, will be displayed in the user interface. However, the Feature Index only stores feature vectors. If the image itself has not been accessed for some time, it might be stored on the slower capacity tier or on the even slower archive tier of the HSM. The information about an expected access is used to trigger migration in the HSM before the result is delivered to the user interface. This mechanism allows loading the image from the HSM while the layout is calculated, and depending on the storage tier may be available together with the exploration result as depicted in Figure 5: Prefetch of semantically relevant data.

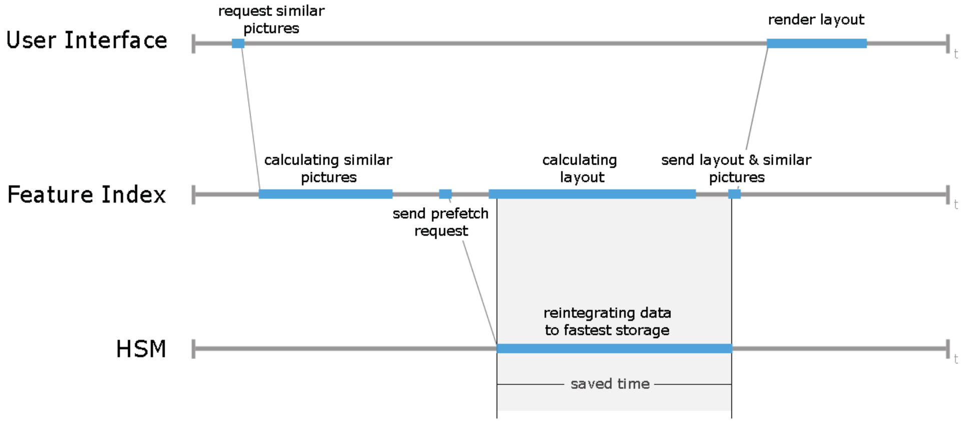


Figure 5: Prefetch of semantically relevant data

A similar optimization can be achieved in the NewsDesk based frontend: Computations based on the currently active widget composition yield information about the possible data types employable by the user interface. If, for example, an image-detail widget is connected to the Graph-Interaction widget, retrieval of the original high resolution image besides the low resolution images used by the graph widget can be triggered to shorten the load time for the detail view as well. More generally spoken, all widget compositions created by the user contain implicit knowledge about the most relevant data types and thus may demand the fastest storage tier for those types.

Knowledge about relationships between entities stored in the HSM and their use in different framework components, such as the user interface or indices, can lead to faster access times. Since the storage component is now able to assess which entities, i.e. files on the storage tiers, are likely to be used together it is able to store and fetch them together, effectively enabling a powerful "prefetch" mechanism based on the meaning of data and interaction.

Besides prefetching data from slower storage tiers the SENSE-Framework is capable of preventing the purge of important data from the fastest storage tiers in the first place. To illustrate that imagine that, in our screnario, the users has found suitable pictures with the help of the exploration tools mentioned above. He now stores some of his findings in a private widget similar to a file system folder, called a lightbox. The domain knowledge provided by the semantic repository of SENSE states that pictures contained in a lightbox are more relevant to the user (and for that more likely to get accessed any time soon) than those who are not and due to that are kept in faster storage tiers as long as possible. If the user, attending the editorial meeting, presents those pictures on a collaborative screen even more users get access to them. This process again has been modeled in the domain dependent ontology to take precedence over pictures stored in a lightbox and accordingly over pictures which are just stored by the system itself.

By that the usage of data within the system leads to a better organized storage structure as well as to better performance in the user interface. We believe that this performance boost cannot be achieved by traditional systems that can only rely on access times and similar, non-semantic meta-information.

The scenario shows how components in the SENSE framework can be connected and work together to increase performance of the overall system. This is due to result processing and retrieval from the HSM system overlap and parallel processing is enabled. This feature, as well as the knowledge about the importance of data to the user is only available due to the integrated semantics about the processes and data stored within the system.

5 Conclusion

In this paper we presented the SENSE framework and how a strong semantic integration from the storage backend up to the user interface can increase performance. We provided a scenario that a system‘s awareness of its components can lead to a shorter time period between a user‘s command and the result delivery and by that improving the usability. We outlined how domain knowledge of a collaborative workflow can be employed to finely graduate the relevance of certain data objects and sketched how this approach can be applied to other domains and applications as well.

The SENSE project is still under way. Extensions and improvements are planned with the near future: The usage of feature index to speed-up and enable interaction schemes will be extended. Currently features based on image-content are used to proof the concept. However, in terms of the project, those indices are far more valuable if they can be used for semantic data such as grouping or entity relations. The ability to quickly discover strong ties between entities can be leveraged in the user interface as well as in the storage tier. We will explore further the possibilities that arise from semantic index integration.

Towards the end of the SENSE project, different studies will be conducted to show an improvement over existing systems, where components have less information about each other. This relates to user experience as well as backend properties like storage throughput and access times.

With a fully functional SENSE framework, additional usage scenarios will be explored and evaluated, leading to novel workflows and novel ways of IT-support for existing workflows.

Achknowledgements

The SENSE project is funded by the Federal Ministry of Education and Research, German Aerospace Center. It is part of the „KMU-Innovativ: IKT" campaign and goes by the funding numbers FKZ 01lS11025A and 01IS11025E.

Litrature Reference

[1] Gantz, J. F.; Chute, C.; Manfrediz, A.; Minton, S.; Reinsel, D.; Schlichtin, W.; Toncheva, A., The Diverse and Exploding Digital Universe, 2008, p. 4

[2] Sire, S.; Vagner, M.; Bogaerts, J., A Messaging API for Inter-Widgets Communication. ACM Proceedings of the 18th international conference on World wide web, 2009

[3] Chudnovskyy, O.; Müller, S.; Gaedke, M., Extending Web Standards-based Widgets towards Inter-Widget Communication, 2012

[4] Pietschmann, S., Radeck, C. and Meißner, K., Semantics-Based Discovery, Selection and Mediation for Presentation-Oriented Mashups. Proceedings of the 5th International Workshop on Web APIs and Service Mashups, ACM ICPS, 2011

[5] Haun, S.; Gossen, T.; Nürnberger, A.; Kötter, T.; Thiel, K.; Berthold, Michael R., On the integration of graph exploration and data analysis – the creative exploration toolkit In: Bisociative knowledge discovery. – Heidelberg [u.a.] : Springer, pp.301–312, 2012

[6] Wehner, P., NewsDesk - Ein hochflexibles, Widget-basiertes Framework für Informationsportale. Proceedings of the GeNeMe'10 conference – Gemeinschaften in Neuen Medien, Technische Universität Dresden, 2010

[7] Faison, T., Event-based Programming: taking events to the limit. Apress, Berkeley, 2006

[8] Fu, W.; Kannampallil T. G.; Kang, R., Facilitating exploratory search by model-based navigational cues. In Proceedings of the 15th international conference on Intelligent user interfaces (IUI ,10). ACM, New York, NY, USA, pp.199–208., 2010

[9] Kasutani, E.; Yamada, A., The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval In: Image Processing, 2001. Proceedings. 2001 International Conference on (Volume:1), pp.674–677, 2001

[10] Radvansky, G.; Zacks, R. T., Mental model organization. Journal of Experimental Psychology: Learning, Memory, and Cognition, 19, pp.95–114, 1991

[11] Brandes, U.; Pich, C., Eigensolver Methods for Progressive Multidimensional Scaling of Large Data. Proc. 14th Intl. Symp. Graph Drawing (GD,06). LNCS 4372, pp.42–53. ©Springer-Verlag, 2007.

[12] Gansner, E. R.; Hu Y., Efficient node overlap removal using a proximity stress model, In 16th Symp. on Graph Drawing, 2008

[13] Tietz, V.; Blichmann, B.; Pietschmann, S.; Meißner, K., Task-Based Recommendation of Mashup Components, Current Trends in Web Engineering, 2011

[14] Wehner, P., Krüger, R.: Semantic-guided communication & composition in a widget/dashboard environment, Proceedings of 13th International Conference on Web Engineering 2013