Journal of Universal Computer Science, vol. 15, no. 8 (2009), 1722-1734
submitted: 17/11/08, accepted: 15/1/09, appeared: 28/4/09 © J.UCS

Fostering Knowledge Flow and Community Engagement in the Development of Interactive Entertainment

Steffen Lohmann, Jörg Niesenhaus, Philipp Heim, Jürgen Ziegler
(University of Duisburg-Essen, Germany
{steffen.lohmann, joerg.niesenhaus, philipp.heim, juergen.ziegler}@uni-due.de)

Abstract: Due to an increasing professionalization, specialization, and globalization in the development of interactive entertainment new demands for comprehensive knowledge management support emerge. This article aims at sensitizing and systematizing the needs and potentials for continuous knowledge flow and community engagement in this application area. It starts with an analysis of typical development activities and involved parties that could benefit from a continuous knowledge management support. Then, a general framework architecture and implementation examples are presented that provide different levels of knowledge management support for interactive entertainment development.

Key Words: knowledge management, interactive entertainment, continuous integration, community engagement, open innovation
Category: M.0, M.1, M.2, M.3, M.7, M.8, D.2.0

1 Introduction

Interactive Entertainment as a major part of the so-called creative industries [DCMS 2001] has gone through an overwhelming economic growth within the past years. As different media is more and more converging and interactive entertainment is becoming an ubiquitous part of everyday life, these trends will be further amplified in the future.

The ongoing growth is accompanied by an increasing professionalization in the development of interactive entertainment. In addition, higher budgets and larger development teams cause a growing specialization. Due to the outsourcing of development activities, the whole process also tends to be more geographically dispersed. Components such as development environments, graphic assets, or technologies are frequently bought from third-parties. Along with this professionalization, specialization, and globalization the documentation and maintenance of knowledge gets a higher priority. Complex dependencies have to be handled and not only data but also knowledge has to be transferred between project partners.

However, though these trends and new demands are increasingly recognized the knowledge management community has shown only little reaction so far. Existing knowledge management solutions have primarily been developed for established markets and are either too monolithic or too specific in order to support highly agile and creative development processes. Appropriate knowledge management solutions for the interactive entertainment industries must ideally satisfy a number of criteria (which we abbreviate as "ALIGN" according to their initials):

- **A**daptable: being easily adaptable to changing project demands
- **L**ightweight: following the principles of simplicity and ease-of-use
- **I**mmediate: adding immediate benefit to the project and all its participants
- **G**eneric: providing a general solution for various projects
- **N**onrestrictive: not dictating strict procedures but fostering creativity

Furthermore, an increasing demand for a stronger involvement of communities that build around interactive entertainment can be observed. Business concepts such as crowdsourcing [Howe 2006] or open innovation [Chesbrough 2003] reflect this growing interest that is both driven by marketing goals (e.g., customer loyalty) and the insight that product users are often highly valuable sources for idea and innovation generation [von Hippel 2005]. These emerging demands must also be addressed by knowledge management solutions aiming to comprehensively support the development process.

2 Potentials for Integrated Knowledge Management

By analyzing typical activities and involved parties in interactive entertainment development, potentials for continuous knowledge management support become apparent.

2.1 Knowledge Flow within the Development Process

At the beginning of the development process in the phases of *pitching* and *pre-production* (see Figure 1) central activities are idea generation and concept creation: Numerous ideas are developed and discarded leading to a permanent change of the product's shape. Converting these agile processes into permanent knowledge is of great value for a development team as it allows to reconstruct at a later time why ideas were discarded, which challenges occurred, and how a problem was finally solved. During development the team occasionally returns to an earlier point of discussion and reconsiders decisions on the basis of a new understanding of the context.

In many cases, previous experiences are included in the considerations so that it is useful to activate knowledge of projects that have been successfully accomplished in the past. Connecting knowledge with project structures, files, and program code enhances the chance of reusing existing components.

With the beginning of the *production* phase the demand for constant documentation increases with every generated product version. In this phase, priority should be given to the interconnection of file structures, data, and knowledge to make every development step traceable and – if required – revocable at a later time.

After completion of a project the review and final feedback discussions start. This phase has come to be called *postmortem* in developer jargon. At this stage the processes, problems, and experiences of the completed project are discussed in order to draw conclusions for future developments. In addition to the feedback given by the developers themselves, the experiences of the service units and publishers as well as feedback of external experts, media representatives, and the community are brought into the discussion.

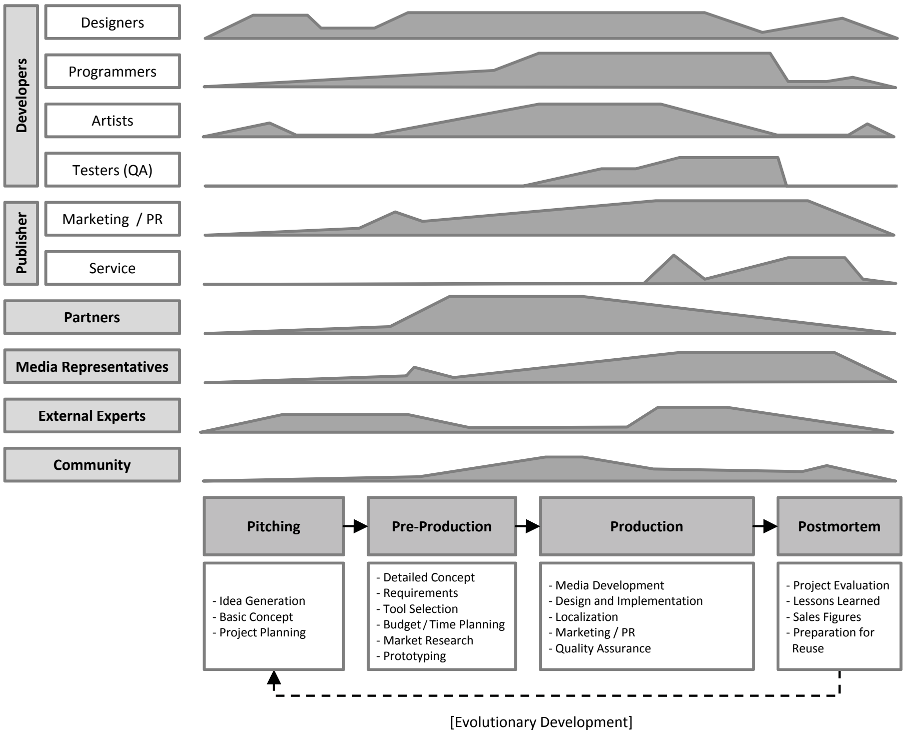


Figure 1: Typical development process in the interactive entertainment domain. The curves describe the distribution of the knowledge flow.

2.2 Knowledge Flow among the Involved Parties

The exact composition of the involved parties is, of course, subject to variation and depends largely on a project's size and goals. Typically, the following groups take part in the development of interactive entertainment (see Figure 1):

The group of *developers* includes all participants actively involved in product development. Typical developers in the interactive entertainment domain are designers, programmers, artists, and testers. All developers contribute with their personal experiences and expertise to the project. Due to an above-average fluctuation of participants in this industry branch it is eminently important to externalize project-relevant knowledge of the individual team members in order to prevent a loss of knowledge when a developer leaves the team.

The *publisher* is responsible for the finance, placement, marketing, and distribution of the product. Besides the continuous dialog with the developers, all knowledge-intensive processes converge at the publisher making it possible to launch a product successfully. The coordination of marketing and public relations, the localization of a product for different markets, and the organization of the distribution are only a few examples of such processes.

Increasing project sizes cause a stronger involvement of outsourcing and offshoring *partners*. This cooperation requires a particularly intensive form of knowledge exchange since it is essential to create a shared understanding of the project goals and to ensure that all externally developed components fit seamlessly into the overall product.

The group of *media representatives* consists of journalists, editors, and producers who work for media formats dealing with interactive entertainment. Often, members of this group get the chance to test an early version of the product in order to prepare previews. The feedback of these previews is of high value for the developers since media representatives are often among the first external persons that review the product. Due to their broad experiences with interactive entertainment they often give valuable advice regarding weaknesses of the tested product version.

Furthermore, *external experts* are involved in the project to assist the developers, for instance, in technical, usability, or child-welfare issues. It is of great importance to the project members to receive early feedback on possible obstacles that may impact the product's success rates or sales numbers. Furthermore, the developers are typically supported by domain experts when designing products for special target groups or application areas.

As mentioned in the introduction, *community* involvement is an increasingly important issue in the development of interactive entertainment. The community that builds around an announced or published product in the interactive entertainment domain is often characterized by high activity and commitment. Communities are a valuable source when it comes to the critical review and discussion of a product or the generation of ideas for improvements and extensions. There are numerous well-established community portals, boards, and weblogs that focus on specific interactive entertainment products or product classes. However, existing methods and tools largely neglect the community as a source for ideas, feedback, and suggestions, though the benefits of an active participation of this group are increasingly recognized.

2.3 Absence of a Global Knowledge Management

So far, no established or even standardized continuous knowledge management support exists for the described development activities and involved parties. Due to the many differences between leisure- and work-related software projects, existing methods and tools cannot be transferred one-to-one.

Normally, development studios use version control systems to administrate data and documents. Knowledge about the interactive product is primarily stored inside the single files of the version control system and is updated at variable time intervals. Outdated knowledge partially exists for any length of time; contrary and inconsistent conclusions across multiple documents are not rare. Usually, definitions about guidelines, design decisions, and responsibilities are spread over several documents and are not maintained in a central location.

Though it is common practice to set references between the documents, this is primarily done manually so that these references often become obsolete or fragmentary as time passes by. In many cases, files and design decisions are insufficiently or not at all commented due to a lack of time. Thus, the function of a file can often only be derived from its location in the file system, the version control structure, and its name. The absence of a global knowledge management is clearly noticeable.

It is primarily not the competitive situation of single team members, teams, or organizational units but rather the additional effort that prevents project participants from externalizing their knowledge. There exists no sufficient support for knowledge documentation that fosters retrieval at a later stage of development or in subsequent projects.

3 Continuous Knowledge Management Support

In order to serve the demands for continuous knowledge management support in interactive entertainment development we propose a framework architecture consisting of a collaboration environment, embedded feedback channels, and knowledge extraction mechanisms. All these components are connected by a central repository that uses semantic technologies for knowledge representation (see Figure 2). In the following, we describe the framework's architecture in more detail and illustrate possible types of support by implementation examples.

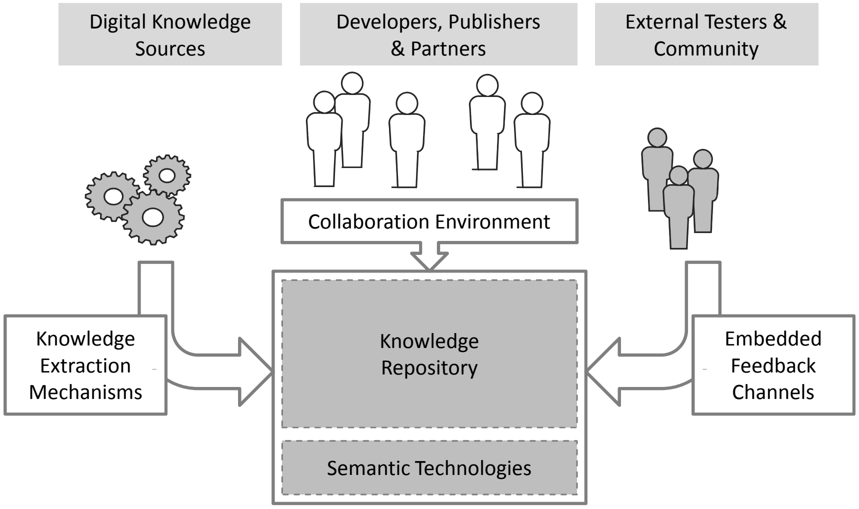

Digital Knowledge Sources
Developers, Publishers & Partners
External Testers & Community
Collaboration Environment
Knowledge Extraction Mechanisms
Knowledge Repository
Semantic Technologies
Embedded Feedback Channels

Figure 2: Framework for supporting continuous knowledge flow in the development of interactive entertainment products

3.1 Knowledge Repository

A knowledge repository forms the central access point for all knowledge management activities in the development process. It stores the project's knowledge in structured form and incorporates the following features:

- *Best Practices knowledge:* Initially, a basic set of ontologies provides conceptualizations that proved to be successful in previous projects. These fundamental structures describe the development project on a rather general level by pointing to important concepts that should be filled with project-specific knowledge during development. The Best Practices knowledge must, however, not only be based on a team's own background. The knowledge repository also enables the reuse of existing ontologies that have been developed in third-party projects or were published specifically for application in software engineering [Torres et al. 2006, Gaevic et al. 2006]. However, the Best Practices knowledge should only be considered as a helpful starting point instead of being misunderstood as a structure the project's knowledge management must adhere to; it should not hamper creativity and innovation.

- *Shared understanding:* The initially provided knowledge base is collaboratively adapted and extended by all project participants during development. It acts as a shared conceptualization that consolidates the different perspectives of the involved parties so that it adequately represents the project's consensual knowledge at any time.

- *Evolutionary conceptualizations:* The knowledge repository is continuously updated in accordance to the project's evolution. History and version control mechanisms allow to track, review, and selectively rollback changes.

- *Context-sensitive integration:* A large part of the knowledge in the repository is semi-automatically derived from the project's context. For instance, if user feedback refers to specific components of the interactive entertainment product (see Section 3.3) a reference to these components is stored along with the feedback in the knowledge repository allowing for future retrieval and reconstruction of the contextual setting. Vice versa, knowledge management support is adapted to the development context ideally providing "the right features at the right time in the right way".

- *Hybrid formality:* The knowledge repository supports different degrees of expressiveness: Some parts of the project's knowledge might already be in a highly structured form while others are less formal and structured. Correspondingly, sophisticated techniques such as automated reasoning can only be applied to parts of a knowledge base that offer sufficient formality.

The knowledge repository is based on semantic technologies, in our case on the XML-based knowledge representation formats RDF, RDF Schema, and OWL [Allemang and Hendler 2008]. By using these Semantic Web standards, ontologies available on the web can easily be added to a knowledge base of the repository. The application of Semantic Web standards is additionally motivated by the fact that the implementation of the framework's components is also mainly based on web technologies in our approach.

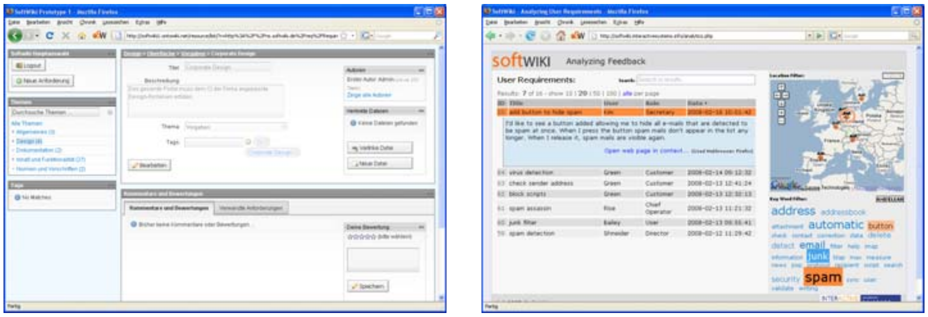
3.2 Collaboration Environment

A collaboration environment provides comprehensive access to the knowledge repository. It is designed according to the principles of simplicity [Maeda 2006] and quick collaboration [Leuf and Cunningham 2001]. Besides the developers, the publishers and partners have separate access rights and are enabled to adapt and update parts of the project's knowledge base. Typical community features such as commenting and rating are combined with semantic technologies allowing for enhanced knowledge retrieval.

The collaboration environment consists of several views, each focusing on a specific user goal. For instance, Figure 3a shows a collaborative structuring and editing view that is based on the *OntoWiki* system [Auer et al. 2006] and has been developed in the context of the SoftWiki project[1]. The user interface provides features for intuitive, web-based editing and updating of knowledge bases

[1] Cooperative research project funded by the German Federal Ministry of Education and Research (BMBF), see http://softwiki.de

and allows easy linking to other knowledge pieces or the underlying topic structure. In addition, participants can 'tag' parts of a knowledge base with freely chosen keywords, resulting in an emerging 'tag space' that represents the participants' vocabulary with respect to the developed product [Riechert and Lohmann 2007]. The effort and formal overhead for expressing knowledge, modifying the knowledge base, or setting relations between knowledge instances is minimized due to the adoption of the Wiki paradigm [Leuf and Cunningham 2001].

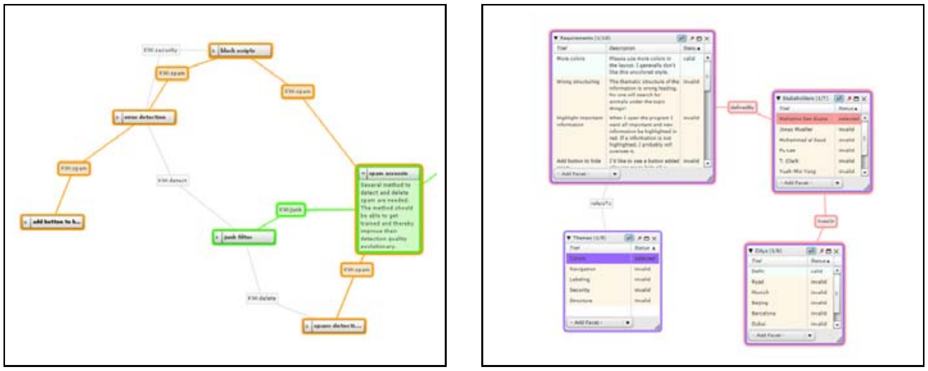


(a) (b)

Figure 3: Two different interfaces of the collaboration environment

Figure 3b shows an alternative view that supports the analysis of feedback gathered by embedded tools (see Section 3.3). A map can be used to filter the feedback according to the geographical location where it has been stated. That way, different markets for an interactive entertainment product can be separately analyzed and evaluated.

In addition to these text- and map-based presentation forms, the collaboration environment needs to provide sophisticated visualizations of the knowledge structure to facilitate understanding and exploratory search. For instance, graph-based visualizations have proven to be useful in visualizing relationships and dependencies in knowledge structures. Therefore, we adapted this type of visualization and enhanced it according to the needs of product development. Figure 4a shows an implementation example that supports the graph-based exploration of a knowledge base along relations of a certain type; all elements that are connected by the selected relation type are arranged in a chain making it easy to follow certain relations [Heim and Lohmann 2009]. Figure 4b shows an alternative visualization that facilitates the exploration of knowledge structures on a conceptual level by combining a graph-based visualization with faceted browsing functionality. By aggregating instances in facets according to the underlying conceptual structure, only relations of instances become visible that are selected in the facets. This prevents the graph-based visualization from getting over-cluttered and facilitates analysis and understanding of the knowledge structure [Heim et al. 2008].



(a)

(b)

Figure 4: Sophisticated visualizations facilitate understanding and exploration of relationships and dependencies

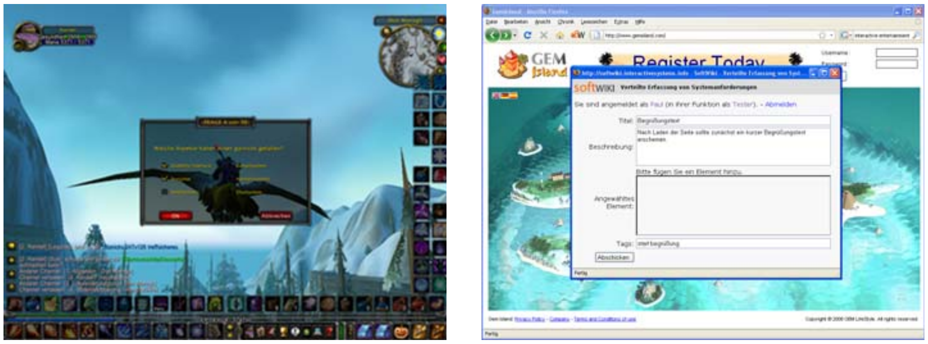
Implementations of the described views on the collaboration environment are in our case based on web technologies enabling access for distributed development teams simply by using a web browser, without the need of installing specific knowledge management software on local devices.

3.3 Embedded Feedback Channels

The central collaboration environment is extended by decentralized feedback channels that can be embedded directly into the development or run-time environments of the interactive entertainment product. Depending on the type of project and state of development, different groups are equipped with appropriate feedback channels (e.g., QA-team, community, external experts).

Figure 5a shows an evaluation tool we implemented in the scripting language LUA [Ierusalimschy et al. 2007]. It can be seamlessly integrated into the run-time environment of the interactive entertainment product – in this case, the online game *World of Warcraft* (WoW)[2] – and enables embedded elicitation of feedback. The tool provides user-initiated input forms that are available at any time in the game and equip the user with an opportunity to report encountered problems or suggestions for improvement. Moreover, it can be used to trigger predefined questionnaires at certain time events or situations.

[2] see http://www.worldofwarcraft.com or http://www.wow-europe.com



softWIKI Verteilte Erfassung von Systemanforderungen

Sie sind angemeldet als Paul (in Ihrer Funktion als Tester). - Abmelden

Titel: Begrüßungstext

Beschreibung: Nach Laden der Seite sollte zunächst ein kurzer Begrüßungstext erscheinen

Bitte fügen Sie ein Element hinzu.

Angewähltes Element:

Tags: startbegrüßung

Abschicken

(a) (b)

Figure 5: Feedback channels embedded in the run-time environments of interactive entertainment products

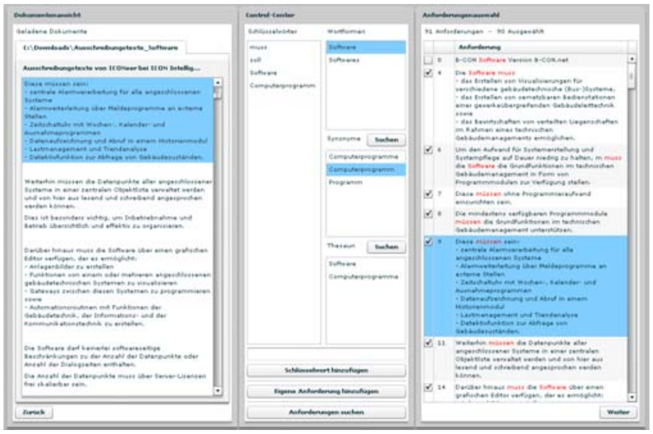
Figure 5b shows an implementation of a feedback channel that can be easily embedded in web browsers in order to elicit feedback on web-based entertainment products. For example, it can be used by the QA team while testing a product version. The testers are enabled to provide feedback directly via their web browser, i.e., without a change of the environment. In addition, the tool captures contextual information by linking feedback to artifacts of the interactive entertainment product or the environment it is used in (e.g., location, time, application status, etc). This contextual knowledge can be highly valuable for later analysis as it allows a more systematic exploration and facilitates the understanding of user feedback.

Integrated feedback channels have a high potential when it comes to foster community engagement. As mentioned in Section 2.2, communities that build around interactive entertainment products are characterized by an above-average activity and commitment. Providing participation opportunities and incentives that stimulate community engagement can be highly valuable to product improvement and innovation generation. With the right tools, communities might be actively involved in interactive entertainment development, leading to products that better meet the users' needs and desires.

3.4 Knowledge Extraction Mechanisms

Next to these forms of knowledge management support requiring active participation of the involved parties, the framework also considers project-related knowledge that is passively provided by available sources. Examples are user statements on weblogs and discussion boards or documents and product descriptions from previous projects. Knowledge available in these sources can be extracted and integrated into the project's knowledge base to get a more comprehensive impression on how an interactive entertainment product is perceived by its consumers.

The integration process follows a semi-automatic approach – manual and automatic activities complement, not replace, each other. The integration must always remain in the control and under supervision of the development team and must not swamp the project's knowledge base with unstructured data.



Diese müssen sein:
- zentrale Alarmverarbeitung für alle angeschlossenen Systeme
- Alarmweiterleitung über Meldeprogramme an externe Stellen
- Zeitschaltuhr mit Wochen-, Kalender- und Ausnahmeprogrammen
- Datenaufzeichnung und Abruf in einem Historienmodul
- Lastmanagement und Trendanalyse
- Detektivfunktion zur Abfrage von Gebäudezuständen.

Weiterhin müssen die Datenpunkte aller angeschlossenen Systeme in einer zentralen Objektliste verwaltet werden und von hier aus lesend und schreibend angesprochen werden können.

Dies ist besonders wichtig, um Inbetriebnahme und Betrieb übersichtlich und effektiv zu organisieren.

Darüber hinaus muss die Software über einen grafischen Editor verfügen, der es ermöglicht:
- Anlagenbilder zu erstellen
- Funktionen von einem oder mehreren angeschlossenen gebäudetechnischen Systemen zu visualisieren
- Gateways zwischen diesen Systemen zu programmieren sowie
- Automationsroutinen mit Funktionen der Gebäudetechnik, der Informations- und der Kommunikationstechnik zu erstellen.

Die Software darf keinerlei softwareseitige Beschränkungen zu der Anzahl der Datenpunkte oder Anzahl der Dialogseiten enthalten.

Die Anzahl der Datenpunkte muss über Server-Lizenzen frei skalierbar sein.

Schlüsselwörter	Wortformen
muss	Software
soll	Softwares
Software	
Computerprogramm	

Synonyme: Computerprogramme, Computerprogramm, Programm

Thesauri: Software, Computerprogramme

91 Anforderungen – 90 Ausgewählt

		Anforderung
☐	0	B-CON Software Version B-CON.net
☑	4	Die Software muss - das Erstellen von Visualisierungen für verschiedene gebäudetechnische (Bus-)Systeme, - das Erstellen von vernetzbaren Bedienstationen einer gewerkeübergreifenden Gebäudeleittechnik sowie - das Bewirtschaften von verteilten Liegenschaften im Rahmen eines technischen Gebäudemanagements ermöglichen.
☑	6	Um den Aufwand für Systemerstellung und Systempflege auf Dauer niedrig zu halten, m muss die Software die Grundfunktionen im technischen Gebäudemanagement in Form von Programmmodulen zur Verfügung stellen.
☑	7	Diese müssen ohne Programmieraufwand einzurichten sein.
☑	8	Die mindestens verfügbaren Programmmodule müssen die Grundfunktionen im technischen Gebäudemanagement unterstützen.
☑	9	Diese müssen sein: - zentrale Alarmverarbeitung für alle angeschlossenen Systeme - Alarmweiterleitung über Meldeprogramme an externe Stellen - Zeitschaltuhr mit Wochen-, Kalender- und Ausnahmeprogrammen - Datenaufzeichnung und Abruf in einem Historienmodul - Lastmanagement und Trendanalyse - Detektivfunktion zur Abfrage von Gebäudezuständen.
☑	11	Weiterhin müssen die Datenpunkte aller angeschlossener Systeme in einer zentralen Objektliste verwaltet werden und von hier aus lesend und schreibend angesprochen werden können.
☑	14	Darüber hinaus muss die Software über einen grafischen Editor verfügen, der es ermöglicht:

Figure 6: Tool supporting knowledge extraction

With the *Semantic Integrator* (see Figure 6) we proposed a tool that supports semi-automatic knowledge discovery in large datasets and integration into an existing ontological structure [Lohmann et al. 2007]. Document sources can be mined for project-related contents by composing search queries with relevant concepts from the project's knowledge base. The results are presented in structured form; project-related terms and paragraphs are highlighted. Statements that are considered as relevant for the project can be extracted and integrated into the knowledge base according to its conceptual structure (e.g., as feature requests, ideas for improvement, etc).

3.5 Syndication and Dissemination

Finally, the knowledge repository provides interfaces for syndication and further processing of parts of a knowledge base (e.g., via web services or news feeds). That way, a developer weblog or marketing newsletter can easily be connected. The other way around, external knowledge (e.g., provided by hardware producers) can also easily be integrated via appropriate interfaces.

4 Conclusion

We tried to point out in this article that continuous integration of knowledge management support in the development process of interactive entertainment is not only crucial for the success of large and distributed projects but also results in several benefits for the participants. These include easier adherence to the timetable and lower dependency on the knowledge of individuals reducing the risks and costs of development. Furthermore, continuous knowledge management facilitates the development of series and secondary or downstream exploitation.

We analyzed and systematized typical knowledge-intensive activities and involved parties and proposed a general framework architecture aiming to serve the demands of interactive entertainment development. Our overall goal was to take a first step towards a better support for knowledge flow and community engagement in agile and creative development processes. However, it has become clear that this goal is faced with several unique challenges. Thus, combined efforts are needed in order to realize the vision of a continuous knowledge management support for the interactive entertainment domain.

Acknowledgements

This work was partly funded by the German Federal Ministry of Education and Research (BMBF) under project no. 01ISF02C. We thank all project partners for the valuable discussions and Timo Stegemann for the implementation work on the Semantic Integrator.

References

[Allemang and Hendler 2008] Allemang, D., Hendler, J.A.: "Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL"; Morgan Kaufmann (2008).

[Auer et al. 2006] Auer, S., Dietzold, S., Riechert T.: "OntoWiki – A Tool for Social, Semantic Collaboration"; Proceedings of 5th International Semantic Web Conference, LNCS 4273, Springer (2006), 736-749.

[Chesbrough 2003] Chesbrough, H.: "Open Innovation: The New Imperative for Creating and Profiting from Technology"; Harvard Business School Press, Boston (2003).

[DCMS 2001] Department of Culture, Media and Sport, UK: "Creative Industries Mapping Document 2001 (2 ed.)"; http://www.culture.gov.uk/reference_library/publications/4632.aspx, retrieved on 2008-11-22.

[Feldt 2007] Feldt, K.C.: "Programming Firefox: Building Rich Internet Applications with XUL"; O'Reilly Media (2007).

[Gaevic et al. 2006] Gaevic, D., Djuric, D., Devedic, V.: "Model Driven Architecture and Ontology Development"; Springer (2006).

[Heim et al. 2008] Heim, P., Ziegler, J., Lohmann, S.: "gFacet: A Browser for the Web of Data"; Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW), (2008), 49-58.

[Heim and Lohmann 2009] Heim, P., Lohmann, S.: "ChainGraph: A New Approach to Visualize Shared Properties in Resource Collections"; Proceedings of the 9th International Conference on Knowledge Management and Knowledge Technologies, to appear.

[von Hippel 2005] von Hippel, E.: "Democratizing Innovation"; MIT Press (2005).

[Howe 2006] Howe, J.: "The Rise of Crowdsourcing"; Wired, 14.06 (2006).

[Ierusalimschy et al. 2007] Ierusalimschy, R., Figueiredo, L.H.d., Celes, W.: "The Evolution of LUA"; Proceedings of the 3rd ACM SIGPLAN Conference on History of Programming Languages, ACM (2007).

[Leuf and Cunningham 2001] Leuf, B., Cunningham, W.: "The Wiki Way: Quick Collaboration on the Web"; Addison-Wesley Longman (2001).

[Lohmann et al. 2007] Lohmann, S., Heim, P., Ziegler, J.: "Semantic Integrator: Semi-Automatically Enhancing Social Semantic Web Environments"; The Social Semantic Web 2007 – Proceedings of the 1st Conference on Social Semantic Web (CSSW), LNI 113, Kllen (2007), 167-172.

[Maeda 2006] Maeda, J.: "The Laws of Simplicity"; The MIT Press (2006).

[Riechert and Lohmann 2007] Riechert, T., Lohmann, S.: "Mapping Cognitive Models to Social Semantic Spaces – Collaborative Development of Project Ontologies"; The Social Semantic Web 2007 – Proceedings of the 1st Conference on Social Semantic Web (CSSW), LNI 113, Kllen (2007), 91-98.

[Torres et al. 2006] Torres, F.R., Calero, C., Piattini, M. (eds.): "Ontologies for Software Engineering and Technology"; Springer (2006)