Marrying Game Development with Knowledge Management: Challenges and Potentials

Jörg Niesenhaus and Steffen Lohmann

Abstract. The game industry has long been neglected as a market and research area for knowledge management and semantic technologies. However, as the budgets for game projects are growing and game development is subject to an increasing professionalization and specialization coupled with outsourcing and offshoring, new needs and potentials for continuous knowledge management and the use of semantic technologies emerge. This chapter starts with a description of the current situation and examines typical game development activities and involved parties that could benefit from a continuous knowledge management support. Subsequently, it provides a general framework architecture and implementation examples that show how knowledge management and semantic technologies can be employed to support game development.

1 Game Development as an Application Area for Knowledge Management and Semantic Technologies

The game industry has gone through an overwhelming economic growth within the past years and analysts foresee a strong growth in the nearby future as well. The branch of game development and publishing is already a major industry with its strongest markets in North America, Japan, and Europe. In 2007 the nine European core markets[1] sold video and computer games worth € 7.3 billion (excluding hardware sales) [15]. Games software sales in the U.S. recorded € 6.9 billion (9.5 billion

Jörg Niesenhaus and Steffen Lohmann
University of Duisburg-Essen
Interactive Systems and Interaction Design
Lotharstrasse 65, 47057 Duisburg, Germany
e-mail: {joerg.niesenhaus,steffen.lohmann}@uni-due.de

[1] The European core markets are Great Britain, Germany, France, Italy, Spain, the Netherlands, Switzerland, Sweden, and Finland.

S. Schaffert et al. (Eds.): Networked Knowledge - Networked Media, SCI 221, pp. 321–336.
springerlink.com

dollar) in 2007 and the Asia-Pacific market earned € 7.4 billion in 2006[2]. The industry's annual growth ranges from 17 percent (U.S. sales between 2003 and 2006) [7] up to 21 percent (German sales between 2006 and 2007) [4]. PricewaterhouseCoopers predicts a doubling of the German market size of 2006 for computer and video games reaching € 2.6 billion in 2010 [4]. One reason for this steady growth is the diversification of gaming products converting more and more casual gamers into active consumers[3]. Furthermore, people that were socialized with games often tend to keep on gaming resulting in an age above 30 years for the average American or British gamer[4] [15].

1.1 Application Area for Knowledge Management

The ongoing economic growth is accompanied by an increasing professionalization in the development of digital games. In addition, higher budgets and larger development teams cause a growing specialization. Large scale projects involve up to 1,000 participants nowadays and accumulate development costs up to tens of millions of euros[5]. Due to the outsourcing of development parts, the whole process also tends to be more geographically dispersed. Specific components (e.g., game engines, graphic assets, level artifacts, or development environments) are frequently bought from third-party developers [20].

Along with this professionalization, specialization, and globalizaton, the documentation and maintenance of knowledge in game projects gets a higher priority. Complex dependencies have to be handled and not only data but also knowledge has to be transferred between project partners. Furthermore, a huge amount of knowledge on game-specific aspects is generated, such as the design of the game world, the storytelling, the basic game mechanics, and technical specifications (cp. [19]). In simple terms:

> Nowadays, an essential part of game development is knowledge management.

However, any attempt that aims to support knowledge management in the development of digital games is faced with specific challenges as the development activities are characterized by highly agile and creative processes. Thus, appropriate

[2] Official numbers of 2007 were not available at time of writing.

[3] Popular examples for this diversification are music games such as "Singstar" (a popular competitive karaoke game by Sony) and "Guitar Hero" (a video game that is delivered with plastic guitar controllers enabling gamers to playback famous songs) or "brain training" games (compilations of several math games and puzzles which are meant to train the human brain) such as "Brain Age" or "Big Brain Academy" as well as intuitive control schemes offered by modern consoles.

[4] According to the Entertainment Software Association (ESA).

[5] An example for a large scale project is *Grand Theft Auto IV* by Rockstar Games [22].

knowledge management solutions must satisfy a number of criteria (which we abbreviate with the term "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

1.2 Application Area for Semantic Technologies

The complex dependencies that connect the artifacts involved in game development and the knowledge structure of the whole project can benefit from a semantic grounding. In general, semantic technologies attracted significant attention in software engineering recently. Initiatives such as the W3C's Ontology Driven Architecture (ODA) [24] or the OMG's Ontology Working Group [25] testify the growing interest for ontology-based approaches in software engineering and related disciplines. Furthermore, some lightweight approaches connecting the so-called Social and Semantic Web such as Semantic Wikis [21] or Social Semantic Annotation [9] are promising candidates for application in game development, since they are normally easy to use, require only little effort and can simultaneously provide a semantic structure that is of great help in large software development projects [18].

However, not only the development process can benefit from the application of semantic technologies. As game worlds are often large in size and complex in their dependencies, they are normally represented in huge class hierarchies or ontology-like strutures. Authentic storytelling or the behavior of non-player characters (NPCs) are typical cases where the game logic uses rules and inferences upon a formal representation of the game world. Research in the area of interactive digital storytelling has shown many opportunities to organize huge amounts of data in order to create believable story worlds. For instance, Crawford suggests an "inverse parser", a dramatic sublanguage for storytelling applications where users select predefined words and terms to communicate with the story world [6]. As a basis for the sublanguage Crawford proposes the use of systems like WordNet[6]. Another approach in this area is the system "HEFTI"[7] by Ong and Legett that uses a knowledge base as an combination of "story components" that represent certain time segments in the story and are composed by a "story builder" instance [16]. The knowledge base also includes "contextual sets" to categorize different plot entities, events, scripts, actions, and characters within the story components.

Other approaches analyze games on a more general level. For instance, the aim of the "Game Ontology Project" [26] is the development of an ontology that

[6] *WordNet* is a lexical database which groups substantives, verbs, adjectives and adverbs into sets of cognitive synonyms, each expressing a distinctive concept; `http://wordnet.princeton.edu/`

[7] *HEFTI* stands for Hybrid Evolutionary-Fuzzy Time-based Interactive Storytelling System.

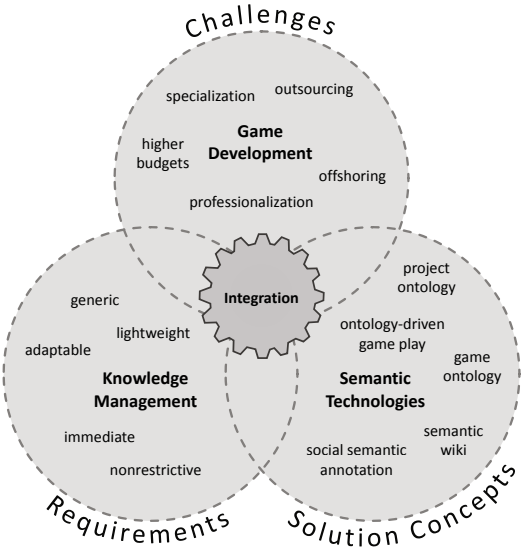

Fig. 1 Challenges, requirements, and promising solution concepts in the integration of game development, knowledge management, and semantic technologies

describes central structural elements of digital games and their interrelations. All these approaches reflect the growing demands for semantic technologies to formally describe games and game worlds and generate intelligent character behavior and authentic storytelling.

Figure 1 summarizes the challenges, requirements, and potential solution concepts regarding the integration of game development, knowledge management, and semantic technologies. In the following, we focus on the support of knowledge-intensive activities in the game development process as a promising application area of semantic technologies. Based on a description of the state of the art in Section 2, potentials for integrated knowledge management are identified in section 3. In Section 4 we propose a framework aiming to support knowledge management in game development on three distinct levels. We close with a conclusion and outlook on future work in Section 5.

2 State of the Art

So far, no established or even standardized continuous support for qualitative knowledge management exists in the area of digital games. Due to many differences between game projects and work-related software projects, established methods and tools from software engineering cannot be transferred one-to-one. Tools currently

used by game development teams are mostly designed for single use cases and are combined by the teams in order to support more complex tasks.

2.1 Data and Document Management

Normally, development studios use version control systems (such as *Perforce*[8], *Alienbrain*[9]) in order to administrate data and documents. Knowledge about technical and game mechanical issues is primarily stored inside the single files of these version control systems that are updated in variable time intervals. Outdated knowledge partially exists for any length of time; contrary and inconsistent conclusions across several documents are not rare.

Usually, it is differentiated between runtime files and design documents. The game is generated out of the runtime files (e.g., engine, scripts, graphics) in its different design stages. The design documents describe guidelines and design decisions concerning the use of technology and game mechanics and define responsibilities. Though it is common practice to set references between the documents, this is primarily done manually so that references often become obsolete or fragmentary as time passes by. In many cases, most files 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 in this structure.

In addition to the version control systems, web-based project management platforms (such as *Mantis*[10], *Jira*[11]) are used to support, for instance, bugtracking, task coordination, and change requests. Due to the lack of a comprehensive platform solution for game development, some studios and publishers develop their tools in order to customize the workflows according to the needs of their project.

It is essential for the project to establish well-defined workflows and knowledge exchange policies that are easy to handle and not too time-consuming. Another vital point is the establishing of a project vocabulary or use case examples in order to give team members an idea on how to contribute to the system and to ensure a consistent form of knowledge storage.

2.2 Outsourcing Content and Code Production

Outsourcing content and code production in game projects places special demands on knowledge management. External teams and partners have to match the existing

[8] *Perforce* is a software configuration management system; `http://www.perforce.com`

[9] *Alienbrain* is an asset management system supporting graphic intensive projects; `http://www.softimage.com/products/alienbrain`

[10] *Mantis* is a web-based bugtracking system; `http://www.mantisbt.org/`

[11] *Jira* is a browser-based bugtracking system with customizable workflows; `http://www.atlassian.com/software/jira/`

production standard and integrate their code pieces, graphic assets, and further multimedia objects into the given game structure.

In order to share data with external partners, development studios often use web-based document sharing platforms (such as *Microsoft Sharepoint*[12]). In addition, specific communication channels protected by security protocols (e.g., Virtual Private Network (VPN)) provide connections for critical data transfers between partners. Further information is often communicated via e-mail or the above-mentioned web-based project management platforms.

Given this infrastructure, the transfer of data is well supported but when it comes to communicate knowledge between the commissioning studio and external partners problems start to emerge. Loose design documents and knowledge scattered across several e-mails or postings on web-based platforms result in inaccuracies and misconceptions in the development process which are time-consuming for both sides. The commissioning studio outsources development tasks with the goal of saving manpower and money but the additional communication and coordination overhead is often underestimated and not considered in the calculations.

2.3 Media Exchange and Consistent Game Design

The realization and communication of a consistent game design is not only challenging for the cooperation between commissioning studios and external partners but also for the studio's internal communication. Much of a game's knowledge is implicitly provided by artworks, videos, mockups, and other media formats. To ensure a consistent design, it is therefore important to set up relations between these different media artifacts and to provide sufficient metadata. Since the development of the game design is a creative and complex process that normally involves several departments of a development studio, the communication of technical and artistic aspects is nearly equally important. Metadata extensions for graphic files (such as XMP [1]) are a first step in the direction of coupling data and knowledge exchange. However, metadata is yet not widely used within game development. To sum up: Developing a consistent game design and communicating it to the project participants, particularly to outsourcing partners, is a crucial task for the success of a game that is not sufficiently supported at the moment.

3 Potentials for Integrated Knowledge Management

By analyzing typical activities and involved parties in the development of digital games, the potentials for continuous knowledge management support become apparent.

[12] *Microsoft Sharepoint* is a browser-based collaboration and document management platform; `http://office.microsoft.com/en-us/sharepointserver/default.aspx`

3.1 Knowledge Management with Respect to the Development Process

At the beginning of the game development process in the phases of *pitching* and *pre-production* (see Figure 2) central activities are the generation of ideas and the creation of new game concepts: Numerous ideas are developed and discarded leading to a permanent change of the game'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 of a game the team occasionally returns to an earlier point of discussion and reconsiders decisions on the basis of a new understanding of the game context. In many cases, previous experiences are included in the considerations so that it is furthermore useful to activate knowledge of projects that have been successfully accomplished in the past. Knowledge should always be connected with project structures, files, and program code to enhance the chance of reusing already existing components [19].

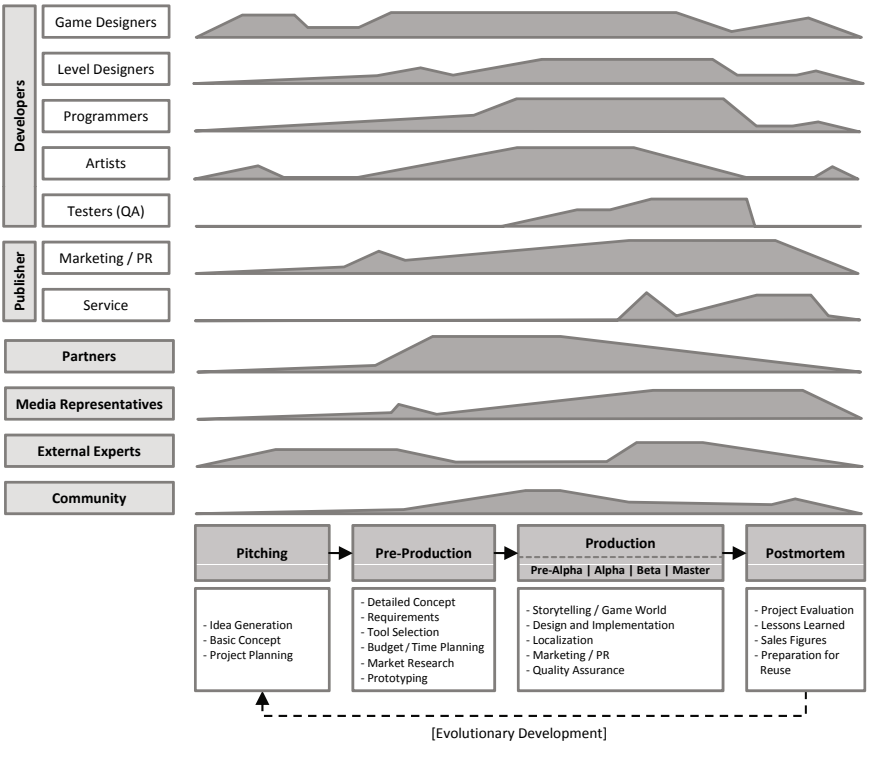

Fig. 2 Typical game development process. The curves describe the distribution of knowledge-intensive activities.

With the beginning of the *production* phase, the demand for constant documentation increases with every generated version of the game. 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. The production phase is usually initiated by the core team of the development studio that sets up the technical infrastructure of the project, such as the engine technology and development environment, and defines the fundamental game design. Then, the team is gradually expanded by members of the artistic and programming departments. In later production phases specialists such as story writers, sound and level designers, voice actors, or testers complement the team. Being not able to share all knowledge right from the start of the project, it is apparent that people who join the team at a later development stage need to quickly catch up all relevant knowledge about the game.

After the completion of a game 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, the experiences of the service units and publishers as well as feedback of external experts are brought into discussion. Furthermore, reviews of magazines and community feedback are discussed in order to identify critical issues that need to be patched immediately or in a future version of the game. It is differentiated between features that were very well received by the community and features that earned a lot of criticism. Afterwards, the criticized aspects of the game are analyzed in detail and the team decides whether to invest time and resources in the improvement of these aspects or to skip them. The outcomes of this review and discussion phase define the basis for subsequent developments and should therefore be saved in structured form allowing easy access in the future.

3.2 Knowledge Management with Respect to 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 digital games (see Figure 2):

The group of *developers* includes all participants actively involved in product development, such as game designers, programmers, graphic artists, level designers, etc. All developers contribute with their personal experiences and expertise to the project. Due to an above-average fluctuation of participants between development teams or departments it is eminently important to externalize project-relevant knowledge of the individual team members in order to prevent a loss of knowledge when a member leaves the team.

The *publisher* is responsible for the finance, placement, marketing, and distribution of the game. Besides the continuous dialogue 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 game for different markets, and the organization of the distribution are only a few examples for these processes. Publishers often distribute several products at a time; therefore, it is vital for them to know when a product is ready to ship in order to calculate the costs, start the marketing campaigns, sign contracts with the retailers and present the products at trade fairs. If problems occur and a project fails to match the gold master date[13] the publisher has to update the planning and contracts. Based on the given information, publishers decide on additional human or financial resources for the project to finish it as fast as possible or they cancel it if the success of the project is doubtful or if it can be foreseen that an project extension will consume far too many resources.

The cooperation with outsourcing and offshoring *partners* requires a particularly intensive form of knowledge exchange since it is essential to create a shared understanding of the project and to ensure that all externally developed components fit seamlessly into the game (see Section 2.2). Typical outsourcing partners are studios or freelancers that specialized on graphical assets, sound design, and voice acting or specific game contents such as level design or artificial intelligence programming. Some of the tools that have been developed by the studio are also used by partners. Therefore, it is important to share not only updates or new versions of a tool but additionally all relevant knowledge about it (e.g., bug reports, new features, how-tos, etc.).

The group of *media representatives* consists of journalists, editors, and producers who work for media formats dealing with digital games. Often, members of this group get the chance to test an early version of the game in order to prepare previews. The feedback of these previews is of great help for the developers since media representatives are often among the first external persons that review the game. Due to their broad experiences with digital games they often give valuable advice regarding bugs or weaknesses of the tested version. Moreover, journalists have to match their own deadlines before magazines go to press or television reports are produced; therefore, it is important for publishers and development studios to know these deadlines when releasing news, demo versions, or media of an upcoming game that shall have a specific media coverage (e.g., a cover page on a magazine).

Furthermore, *external experts* are often involved in game projects to assist the developers, for instance, in technical, usability, or child-welfare issues. It is of great importance to the developers and publishers to receive early feedback on possible obstacles in order to still have an influence on changes. A high age rating by a public rating agency[14] (e.g., 'Mature' or 'Adult-Only') can have a negative impact on the sales numbers of a game and is an important factor for the calculation of a publisher. Frequently, developers are supported by domain experts when designing products for special target groups or application areas. For instance, pedagogues might support the developers in the areas of serious games [14] and game-based learning [17]. Sometimes, external usability engineers are hired in order to evaluate the game interfaces, identify usability issues or conducting large-scale playability tests [11].

[13] The final version of a game which goes into manufacturing is called *gold master*.

[14] In many countries *public rating agencies* define generally binding age restrictions for media products.

The *community* that builds around published or announced game products is often characterized by a high activity and engagement when it comes to the critical review and discussion of a game or the generation of ideas for improvements and extensions. There are numerous well-established community portals, boards, and weblogs that focus on digital games (e.g., Gamespot[15], IGN[16]). Thus, we consider it valuable to integrate the community as far as possible in the development process in order to gain new ideas, helpful suggestions, and feedback from outside. Some projects go as far as to give away basic decisions of the game design to the community and let the game fans actively participate in the development process (e.g., *Top Secret*[17]). More common, some game developers offer polls or questionnaires to the community which might have an impact on certain features or give the developers a better idea of the most wanted features for a game. Another method to involve the community in the development process is to create a "closed beta" (for preselected community members) or a "public beta" (for everyone who is interested) phase where gamers are allowed to play parts of the nearly finished game for free. Closed or public betas are normally combined with ingame or online questionnaires, board discussions, and feedback forms. In addition, the developers evaluate game logs in order to balance features or find bugs.

4 Continuous Integration of Knowledge Management

In order to serve the demands for continuous knowledge management support in game 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 3). In the following, we describe the framework's architecture in more detail and illustrate possible types of support by implementation examples.

4.1 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 project on a rather general level by pointing to

[15] One of the biggest game portals for the American and European market; `http://www.gamespot.com`

[16] Big American and British portal for interactive entertainment and new media; `http://www.ign.com`

[17] *Top Secret* is a massive multiplayer online racing game which is developed under the lead of David Perry together with about 60.000 community members.

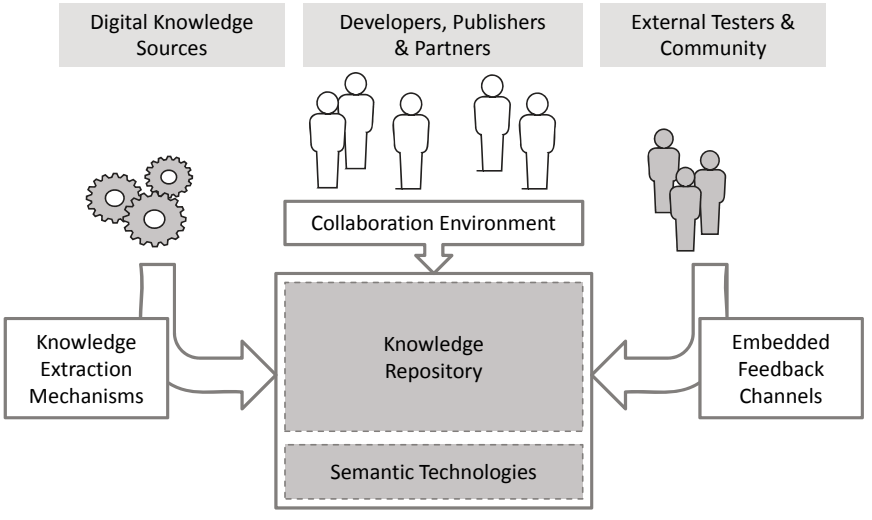

Fig. 3 Framework for supporting knowledge management in game development

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 [23, 8]. 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 game (see Section 4.3) 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, the 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 [3]. By using these Semantic Web standards, ontologies that are 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. This enables access for distributed development teams simply by using a web browser, without the need of installing specific knowledge management software on local devices.

In addition, the knowledge repository provides interfaces for syndication and further processing of parts of the knowledge base (e.g., via web services or news feeds). That way, a developer weblog (see Figure 4a) or mailing list can easily be connected to the repository. The other way around, external knowledge (e.g., provided by hardware producers) can also be integrated via appropriate interfaces according to the access rights.

4.2 Collaboration Environment

A collaboration environment provides comprehensive access to the knowledge repository. It is designed according to the principles of simplicity [13] and quick collaboration [5]. 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 an enhanced knowledge retrieval.

Figure 4b shows an implementation example of the collaboration environment that is based on the *OntoWiki* system [2] and has been developed in the context of the SoftWiki project[18]. The user interface provides features for intuitive, web-based editing and updating of knowledge bases and allows easy interlinking of knowledge pieces or referencing on 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 [18]. The effort and formal overhead of expressing knowledge, modifying the knowledge base, or setting relations between knowledge instances are minimized due to the adoption of the Wiki paradigm [5].

[18] SoftWiki Distributed, End-user Centered Requirements Engineering for Evolutionary Software Development; `http://softwiki.de`

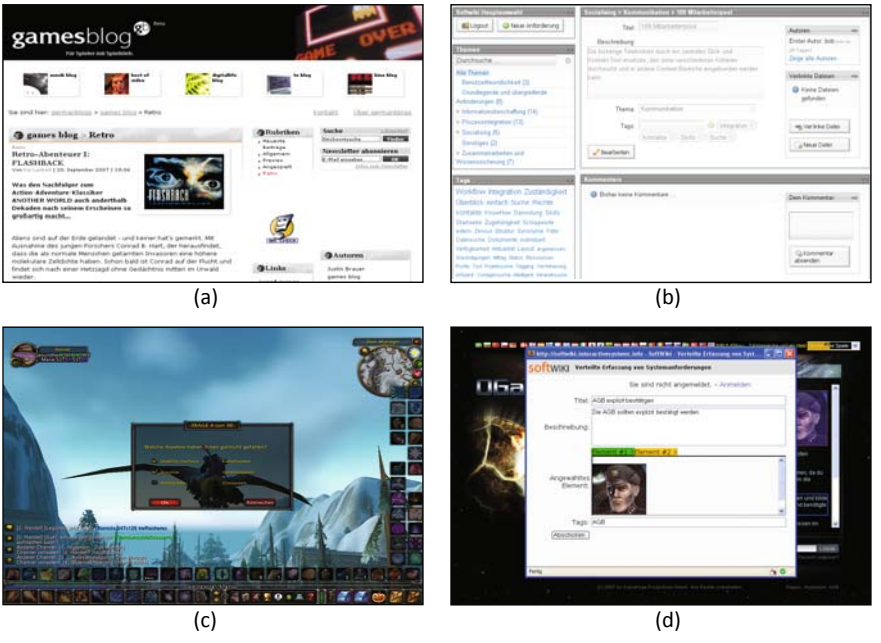

Fig. 4 Examples of knowledge management support for game development

4.3 Embedded Feedback Channels

The central collaboration environment is extended by decentralized feedback channels that can be embedded directly into the development or run-time game environments. 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 4c shows a possible realization implemented in the scripting language LUA [10]. It can be seamlessly integrated into the run-time environment of the game, in this case, of the online game *World of Warcraft* (WoW)[19]. The tool provides user-initiated input forms that are available at any time in the game and equip the WoW 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.

Figure 4d shows an alternative implementation that can be easily embedded in web browsers in order to elicit feedback on browser games without a change of the environment. In addition, the tool captures contextual information by linking feedback to artifacts of the game environment, if possible (e.g., location, time, application status, etc). This contextual knowledge helps in later analysis as it allows for a more systematic exploration and facilitates the understanding of feedback.

[19] `http://www.worldofwarcraft.com` or `http://www.wow-europe.com`

Embedded feedback channels have a high potential when it comes to foster community engagement. As mentioned in Section 3.2, communities that build around games 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 development, leading to games that better meet the users' desires.

4.4 Knowledge Extraction Mechanisms

Next to these forms of knowledge management support requiring active participation of the involved participants, the framework also considers project-related knowledge that is passively provided by available sources. Examples are user statements on weblogs, community portals, and discussion boards or documents and product descriptions from previous projects. Relevant information available in these sources is semi-automatically extracted and integrated into the project's knowledge base in order to get a comprehensive impression on how a game product is perceived by others. Due to an increasing use of XML- and RDF-based formats for content and knowledge representation, automated processing is increasingly feasible.

A central requirement for extraction mechanisms is that the integration process remains always in the control of the developers in order to not swamp the project's knowledge base with unstructured data. With the *Semantic Integrator* we proposed a tool that supports semi-automatic knowledge discovery from large datasets and integration in an existing ontological structure [12]. 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 the conceptual structure (e.g., as feature requests, ideas for improvement, etc).

5 Conclusion

We analyzed and systematized typical knowledge-intensive activities and involved parties in game development and proposed a general knowledge management framework aiming to serve the demands of this application area. In particular, we tried to point out that a continuous integration of knowledge management support and semantic technologies into the development process of digital games is not only crucial for the success of large and distributed projects but also results in several benefits for the involved parties. These include easier adherence to the timetable and lower dependency on the knowledge of individuals reducing the risks and costs of development. Continuous knowledge management support and semantic representation also facilitate the development of game series and secondary or downstream exploitation, for instance, if a similar game structure or the same engine are reused in subsequent

projects. Besides its function as documentation, the knowledge base can also serve as a source of inspiration for these subsequent projects.

Similar to other application areas of software engineering, ontologies can help to create a shared understanding in game projects and are able to support the classification and interoperable exchange of game artifacts and knowledge. The game industry is well-known for being a driver of new technologies. Therefore, game development might be a promising testbed of semantic technologies. Vice versa, semantic technologies and knowledge management solutions that proved to be successful in game contexts might be suitable candidates for application in non-gaming environments. Altogether, the interplay of game development, knowledge management, and semantic technologies offers a lot of potential in different directions that need to be further explored and evaluated. Our general goal was to take a first step towards a continuous knowledge management support for agile and creative development processes. It has become clear that this goal is faced with several specific challenges that require combined research and development efforts.

References

1. Adobe Systems Inc.: Adobe XMP: Adding intelligence to media (2008), `http://www.adobe.com/products/xmp/index.html` (cited December 1, 2008)
2. Auer, S., Dietzold, S., Riechert, T.: OntoWiki – A Tool for Social, Semantic Collaboration. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 736–749. Springer, Heidelberg (2006)
3. Allemang, D., Hendler, J.A.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann, San Francisco (2008)
4. German Trade Association of Interactive Entertainment Software (BUI): Facts (2008), `http://www.biu-online.de/fakten/` (cited December 1, 2008)
5. Leuf, B., Cunningham, W.: The Wiki Way: Quick Collaboration on the Web. Addison-Wesley Longman, Amsterdam (2001)
6. Crawford, C.: Chris Crawford on Interactive Storytelling. New Riders (2005)
7. Entertainment Software Association (ESA): Industry Facts (2008), `http://www.theesa.com/facts/` (cited December 1, 2008)
8. Gaevic, D., Djuric, D., Devedic, V.: Model Driven Architecture and Ontology Development. Springer, Heidelberg (2006)
9. Good, B., Kawas, E., Wilkinson, M.: Bridging the Gap between Social Tagging and Semantic Annotation: E.D. the Entity Describer. Nature Preceedings (2007), hdl: 10101/npre.2007.945.1
10. Ierusalimschy, R., de Figueiredo, L.H., Celes, W.: The Evolution of Lua. In: Proceedings of the 3rd ACM SIGPLAN Conference on History of Programming Languages, ACM, New York (2007)
11. Isbister, K., Schaffer, N.: Game Usability: Advancing the Player Experience. Morgan Kaufmann Press, San Francisco (2008)
12. Lohmann, S., Heim, P., Ziegler, J.: Semantic Integrator: Semi-Automatically Enhancing Social Semantic Web Environments. In: Auer, S., Bizer, C., Müller, C., Zhdanova, A. (eds.) The Social Semantic Web 2007 – Proceedings of the 1st Conference on Social Semantic Web (CSSW). LNI, vol. 113, pp. 167–172. Köllen (2007)

13. Maeda, J.: The Laws of Simplicity. The MIT Press, Cambridge (2006)
14. Michael, D.R., Chen, S.L.: Serious Games: Games that Educate, Train, and Inform. Thomson Course Technology (2005)
15. Nielsen Games Survey: Video Gamers in Europe 2008. Interactive Software Federation of Europe (2008), http://www.isfe-eu.org/tzr/scripts/downloader2.php?filename=T003/F0013/8c/79/w7ol0v3qaghqd4ale6vlpnent&mime=application/pdf&originalname=ISFE_Consumer_Research_2008_Report_final.pdf (cited December 1, 2008)
16. Ong, T., Leggett, J.: A Genetic Algorithm Approach to Interactive Narrative Generation. In: Proceedings of the 15th ACM Conference on Hypertext and Hypermedia. ACM, New York (2004)
17. Prensky, M.: Digital Game-Based Learning. Paragon House Publishers (2007)
18. Riechert, T., Lohmann, S.: Mapping Cognitive Models to Social Semantic Spaces - Collaborative Development of Project Ontologies. In: Auer, S., Bizer, C., Müller, C., Zhdanova, A. (eds.) The Social Semantic Web 2007 - Proceedings of the 1st Conference on Social Semantic Web (CSSW). LNI, vol. 113, pp. 91–98. Köllen (2007)
19. Rollings, A., Morris, D.: Game Architecture and Design: A New Edition. New Riders (2003)
20. Saltzman, M.: Game Creation and Careers: Insider Secrets from Industry Experts. New Riders (2003)
21. Schaffert, S., Bry, F., Baumeister, J., Kiesel, M.: Semantic Wikis. IEEE Software 25(4), 8–11 (2008)
22. Bowditch, G.: Grand Theft Auto producer is Godfather of Gaming. Times Online UK (2008), http://www.timesonline.co.uk/tol/news/uk/scotland/article3821838.ece (cited December 1, 2008)
23. Torres, F.R., Calero, C., Piattini, M. (eds.): Ontologies for Software Engineering and Technology. Springer, Heidelberg (2006)
24. W3C: Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering (2006), http://www.w3.org/2001/sw/BestPractices/SE/ODA/060211/ (cited December 1, 2008)
25. OMG: Ontology Working Group (2008), http://www.omg.org/ontology/ (cited December 1, 2008)
26. Zagal, J., et al.: Towards an Ontological Language for Game Analysis. In: Proceedings of DiGRA 2005 Conference on Changing Views – Worlds in Play (2005)