2009

Enterprise Social Network Platforms as a Managment Tool in Complex Technical Systems

Stefan Bente
Tata Consultancy Services, stefan.bente@tcs.com

Juergen Karla
RWTH Aachen University, karla@winfor.rwth-aachen.de

Enterprise Social Network Platforms as a Management Tool in Complex Technical Systems

Dr. Stefan Bente
Tata Consultancy Services
stefan.bente@tcs.com

Dr. Jürgen Karla
RWTH Aachen University
karla@winfor.rwth-aachen.de

ABSTRACT

Technical systems like communication networks or IT systems tend to grow more and more large and complex. Their management is handled by a workforce divided into different locations and responsibility areas. This adds to the system vulnerability by misconfiguration and hidden side effects of administrative actions. Web 2.0 social network platforms, on the other hand, are a canonical tool to increase information flow and transparency. This paper proposes a custom-tailored social network platform to be used as a tool in managing such systems. A prototype for such a platform has been implemented and, as a thought experiment, applied to the operation support system of a mobile communication network. This experiment shows the strengths of this idea, together with a number of caveats of which one should be aware.

Keywords

Web 2.0, Enterprise 2.0, Social Network Platform, System Management

In recent years, the development of internet services has been summarized under the term Web 2.0. It is based on the changed perception of the World Wide Web (WWW), due to fundamental changes in the integration of the user into the offered services (Tredinnick 2006). The interpretation of the term Web 2.0 differs depending on whether it is regarded from a technical, sociological, or business point of view (Karla 2007). Over the last years, some definitions and comments were given:

- „Web 2.0 is the network as platform, […] consuming and remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects through an 'architecture of participation' […]." (O´Reilly 2005)
- „… a piece of jargon, nobody even knows what it means." (Berners-Lee 2006)
- "…tries to integrate new developments regarding the WWW and to give them a name." (Kaul 2005)

Enterprise social software, also known as Enterprise 2.0, is a term describing social software used in a business context. Enterprise 2.0 is defined as a system of web-based technologies that provide rapid and agile collaboration, information sharing, emergence and integration capabilities in the extended enterprise (AIIM 2008). It includes social and networked modifications to company intranets used by enterprises to organize interactive processes. "Enterprise Web 2.0" describes the introduction and implementation of Web 2.0 technologies within an enterprise, consisting of rich Internet applications and using the web as a general platform.

An integration of Web 2.0 tools into the internal business models and business processes is generally regarded a promising approach for enterprises (Bauer, Mandl 2007). In various analyst reports and conference keynotes, Enterprise 2.0 is ranked amongst the top IT trends (Young 2008; Koplowitz, Young 2007; Eslambolchi 2008). Enterprise usage of Web 2.0 concepts and software can be classified into two categories:

- External (i.e. customer-facing)
- Internal, for example by offering social network tools for the enterprise workplace

The internal usage of social network tools is a strong trend. According to (Young 2008), many CIOs introduce their own Enterprise 2.0 platforms to prevent the employees from forming communities in public platforms like Facebook, with the risk of exposing confidential information there. Evidently there is an intrinsic need in today's complex business world for increased communication between employees, facilitated by easy-to-handle networking tools. Besides, efficiency of processes is still being focused by enterprises.

Forming and supporting communities and connections between members is the goal for social network platforms. They also offer application programming interfaces (API) for applications to be launched into them. One essential element of the platforms is to inform its members about the activities of other members in these applications.

In this paper we propose a custom-tailored social network platform, which uses this principle to manage complex, decentralized technical systems in a better way. An example for such a system is the Operation Support System (OSS) of a mobile network operator, which is used here as a concrete case study. However, it should be kept in mind that the concept presented in this paper is not specific to the telecommunications domain. It can be applied to arbitrary technical systems.

By learning what other users or system operators have done in certain parts of the system, a particular system engineer gains insight in possible problem causes. A prototype for such a platform has been presented in (Bente, Hegle 2008), which we generalize and use in a thought experiment to perform troubleshooting in complex systems.

The subsequent sections describe in detail this concept, the experiment, and the conclusions drawn from it. First, the scope for which the social network platform prototype has been designed is presented. As a concrete example, the management of mobile communication networks is chosen. After that, the social network platform prototype, and the changes applied to the original concept in order to fit it to an enterprise context, are analyzed. Then, a thought experiment using that prototype is conducted, based on three authentic troubleshooting case studies from the mobile network area. Finally, strengths, challenges, and open issues of the approach are summed up, and an outlook to further development is given.

MANAGEMENT CHALLENGES IN COMPLEX TECHNICAL SYSTEMS

Complex decentralized technical systems are manifold in today's globalized world. IT systems, communication networks, manufactories, and distribution systems for energy or water have reached considerable sizes. In addition, these systems are usually interlinked with each other.

On top of the pure operational complexity, such systems undergo a constant renewal process that is triggered by several aspects:

- *Technical*: New machines, technologies, protocols, and IT hardware replace existing infrastructure
- *Architectural*: The system architecture is restructured to be better suited for business, technical, or non-functional requirements (performance, maintenance, or operation cost)
- *Business-driven:* The business model or the product portfolio is transformed
- *Strategic:* Due to company acquisitions, mergers, or carve-outs the technical infrastructure needs to be split or integrated

This infrastructure is managed by a workforce counting in hundreds or sometimes thousands of persons. Usually, the management happens in a distributed fashion, in terms of geographic location, areas of responsibility, and organizational structure (departments, possibly outsourcing).

Taking telecommunication infrastructure as an example, the 20 largest mobile network operators each have more than 50 million subscribers. The largest, China Mobile, even has half a billion customers (China Mobile 2008). These huge networks are usually managed by several network operations centers, often distributed over different regions, time zones, and political entities. In addition, the operating personnel are divided into different areas of responsibility. Engineers monitoring the network resource layer usually belong to a different department than those persons dealing with the service layer. They in turn are separated from the employees operating the IT infrastructure (like databases and servers), or those persons implementing the business rules.

All these factors contribute to the fact that the social structure of the operating personnel for such large and complex systems tends to be large and complex itself. This makes it hard, if not impossible, for an engineer who takes care of a specific logical, physical, and/or regional part of the system, to know and connect to all other operators responsible for dependent system areas.

Looking at typical system faults, the majority of system failures (at least in IT and communication systems, where hardware is very reliable) is caused by (human) misconfiguration or mismanagement of the system. Therefore, especially in such complex systems, the lack of knowledge about other participants in the system management is a substantial handicap in preventing system failures, or at least quickly fixing them. For example, an activity that in itself looks pretty innocent, like renaming a database table supposedly used only by a specific application, may have unforeseen consequences in seemingly unrelated parts of the system.

AN ENTERPRISE SOCIAL NETWORK PLATFORM FOR COMPLEX TECHNICAL SYSTEMS

Figure 1 depicts the a system failure as described above. A large number of persons use the system, or administer it. We assume that one of these activities (or a combination of several activities) has the hidden and involuntary side effect of causing a system failure (depicted by the line at the bottom of the figure).

What kind of tool would help a system engineer to investigate this failure (right side, arrow tagged with a question mark)? If the regular system monitoring doesn't reveal the root cause of the failure, the tool should provide insights into the other users' activities in the system – under the working assumption that these activities might be related to the failure. However, such activity notifications (depicted at the top of Figure 1) need to be filtered by some intelligent algorithm. Otherwise the sheer number of information renders the tool useless.

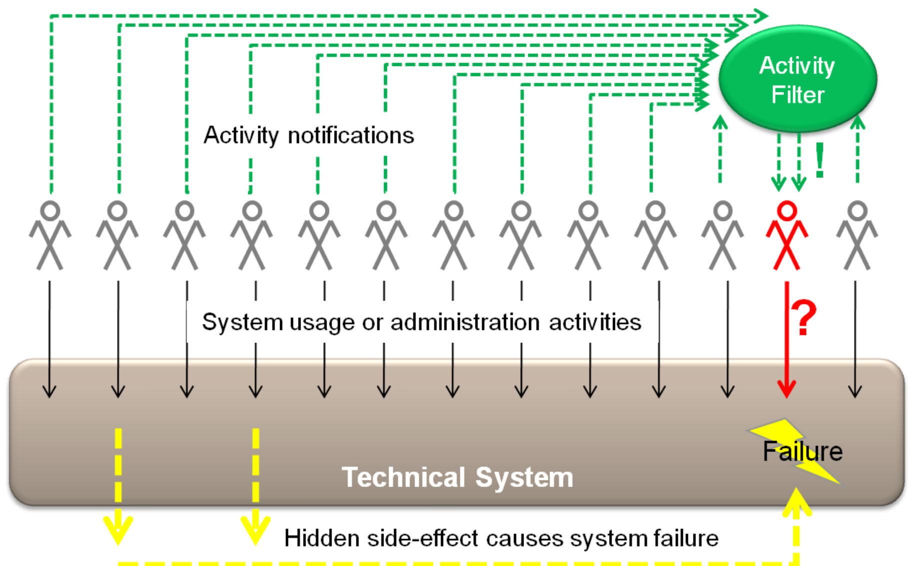


Figure 1: Causes for system failures in complex systems, and role of an "activity filter" to raise awareness

The obvious solution for such a tool is to custom-tailor a social network platform for the purpose at hand. Figure 2 shows main conceptual parts of a "classical" social network platform. The concepts are listed on the right side of the figure, with the font size indicating its weight in the platform.

First of all, there is the "me", being in the middle of a very large crowd of other users of the network. The "me" is connected to a small number of "friends". This is – naturally – the most important concept in the classical social network. It should be noted that "friend" does not necessarily mean "personal friend", rather "someone I share a common interest with". In a job-centered social network like LinkedIn or Xing, only a small fraction of one's contacts (if any) are personal friends. Instead, these are persons to whom a business relation exists. In the broadest sense, "friends" are those network users whose activities are worth tracking.

"Activities" is the next important concept, connected to "applications". In the classical social network, an application is usually relatively small and simple – for example a blog or a movie rating application. Typical platforms offer APIs to deploy new applications to them. Whenever a user performs an activity in an application (like recommending the new James Bond movie), a notification is sent to his friends.

When applying this concept of social network platforms to users of complex technical systems, some changes need to be made. This is depicted in Figure 3. There is still the "me", now in the midst of colleagues, who take the former role of "friends". "Applications" – and this is probably the most severe change of the original concept – are the user interfaces of the technical system itself, for example administrative clients. This means that "activities" are now the actions of the operating personnel, when using or administering the system.

An important difference to the classical platforms is the number of other users. Even for a large technical system, the number of operating users will be in the area of several hundreds, perhaps thousands. This is no match for the millions or users as in commercial public platforms. Foremost, this means that the "friends" concept loses its role as a natural filter for the plethora of activities in the system.

Considering that an active LinkedIn user easily has more than 500 contacts, and that on the other side a technical system may only have a couple of hundred users and operators, a state of saturation can occur. If (nearly) everyone is connected to (nearly) everyone else, notifications of more or less *all* user activities in the system are received. Besides, there are a lot more activities per person than in classical social networks, since the user spends eight hours of concentrated work with the system, not just the occasional half an hour during leisure time.

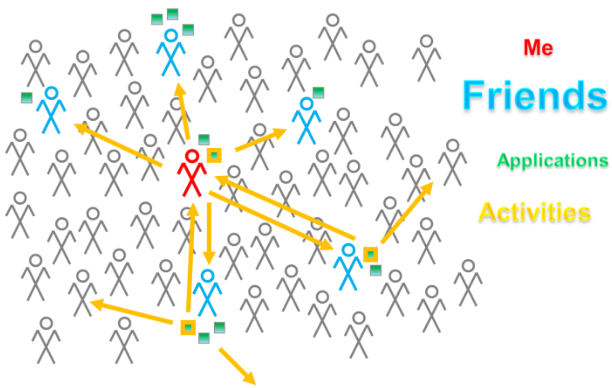


Figure 2: "Classical" social network platform

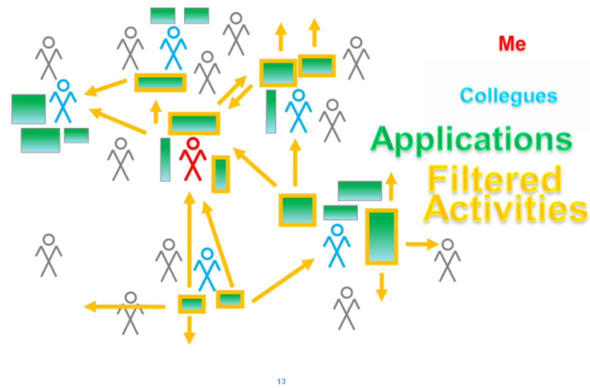


Figure 3: Social network platform for complex technical systems

In addition, the applications are way more complex than the simple applications in public platforms. Therefore, also the activities and their notifications carry a lot of information and contextual references.

The conclusion is that the filtering of activities is very important factor for the usability of such a platform. It requires more sophistication than just selecting only those activities conducted by friends. In fact, friends are not used at all for filtering in our enterprise social network platform; filtering is completely handled by other means, as described below.

Prototype of a Social Network Platform for OSS Systems

In order to test the concept described above, it was applied to a concrete technical field. The technology chosen was an Operation Support System (OSS) for mobile communication networks. Communication networks rank among the most complex technical systems built by human beings. An OSS is responsible for a network area usually the size of a European country, collects operational data from the network, and performs tasks like system monitoring, optimization, planning and so on. Strictly speaking, an OSS is a place where two complex systems meet: first, the managed network with its tens of thousands of distributed network elements, and then the OSS itself[1].

In order to define the activity notifications, it is useful to refer to industry standards of the respective technical domain, so that the integration to the specific tool landscape is not a highly individualized process. For the telecommunications area, the eTOM model (TM Forum 2008) provides a selection of standard processes for communication service providers. Of this process framework, 71 activity types belonging to 28 different eTOM level 3 processes were selected as possible activities logged by the social network platform.

The proposed architecture and integration into an existing OSS landscape is shown in Figure 4. As outlined above, the existing OSS applications are integrated into the platform as "social network applications". When a user performs his standard OSS administration tasks, the applications log activities to the social network platform. This is transparent to the user. He or she can see a filtered view on the activities of his or her colleagues (including his own) in the social network web

[1] To give the reader an impression of the complexity of the OSS itself: The performance management part of an OSS alone receives and processes approx. 1GB of data per hour or more, depending on the network size and type.

client (depicted on the right side of Figure 4). The integration to OSS applications (green boxes on the left side) is performed in an unobtrusive way, that is without changing the application[2]. Integration interface is the eTOM model.

Of the complete architecture shown in the figure below, only the web client and the integration interface has been specified and implemented so far. The processing and persistance logic is yet missing (although this part of the system can be built with standard software components, and holds no novelty value in itself).

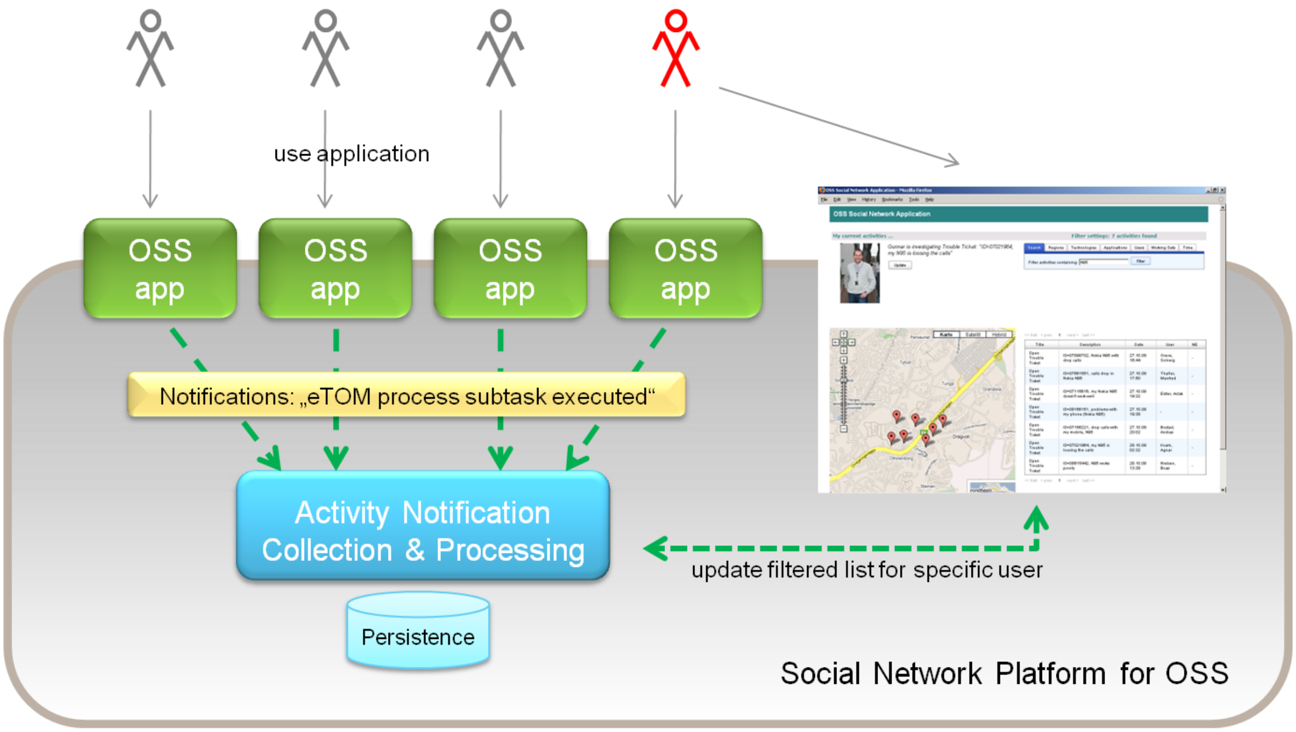


Figure 4: Integration of the OSS Social Network Platform into an OSS Landscape

The data model for the activity notification is kept as simple as possible, containing the attributes listed below. The user of the social network can filter the incoming notifications according to these fields.

- Title, description
- eTOM processes subtask
- Geographic reference
- User
- Technology reference (GSM, Transport, 3G, ...)
- Application reference

THOUGHT EXPERIMENT: USING THE SOCIAL NETWORK PROTOTYPE TO TROUBLESHOOT AN OSS SYSTEM

The applicability of social networks for typical mobile network management situations was tested in a thought experiment using the prototype described above. This approach allowed it to first test the usefulness of the concept for this particular problem domain, without having to perform an actual integration into the OSS landscape up-front.

The experiment was conducted in collaboration with the Norwegian mobile network operator (MNO) NetCom A.S., a subsidiary of TeliaSonera. TeliaSonera is the world's sixth largest MNO, with its main customer base in the northern European countries (GSMA 2008).

[2] Options are: Usage of server side application APIs, database triggers, and for web-based application clients injection of JavaScript code or usage of browser plugins.

The experiment is set in Norway. The mobile network is a virtual one, modeled closely according to the real network of NetCom as a mixed 2G/3G network. 3G coverage is available only in urban areas like Oslo, Trondheim, Bergen and so on, while 2G coverage is available for the whole of Norway, also in the sparsely populated countryside. For the thought experiment, roughly 260 locations in mid Norway were selected as likely positions for network elements. The virtual locations were sanity-checked by NetCom, and reflect a real network setup with sufficient approximation. In addition, roughly 460 customer locations were randomly picked.

Based on the experiences of NetCom with the management of their network, a random algorithm generated incidents from the pool of activity types modeled in the OSS Social Network prototype, using the virtual network element and customer locations in Norway. These activities are conducted by 60 virtual employees of NetCom.

All in all, this generated a cloud of approx. 600 activities for a period of 30 days. The activity cloud was used to obfuscate notifications pointing to hidden cause-effect relationships, and to test the prototype's filtering concept. This pragmatic number was chosen in order to use the prototype for experimenting without worrying too much about performance optimization. It can be assumed that, given a complete integration of the Enterprise Social Network in a major OSS management application, the number of activities will be higher at least by one order of magnitude.

Example 1: Misconfiguration of Handover Parameters for a Cell

Handover is a part of the 2G/3G protocol that allows a mobile device to change its location over cell boundaries while being in use (e.g. if a user drives his car during a phone call). The technical procedure behind this is relatively complex, and needs to be configured through specific network element parameters.

If these parameters are misconfigured, the handover fails, resulting in a dropped call. If there is only a slight flaw in the configuration, the handover fails only for certain cases, for example for a certain cell phone type. Other phones are correctly handed over to the neighboring cell.

Such a problem is hard to detect using the typical OSS assurance tools, because the absolute number of handover failures is relatively small, since only a few cell phones are concerned. Therefore the network performance degradation will show up only slowly (if at all). Nonetheless, the service problem really exists, since a user owning such a phone will face the same dropping calls over and over again, whenever he or she is moving through the affected network cell.

For the scenario, we assume that such an angry phone user calls the level 1 support and reports his or her constant problems with dropping calls near the home zone. The level 1 support escalates the problem further to level 2, then to level 3, where a network engineer named Gunnar tries to resolve the issue.

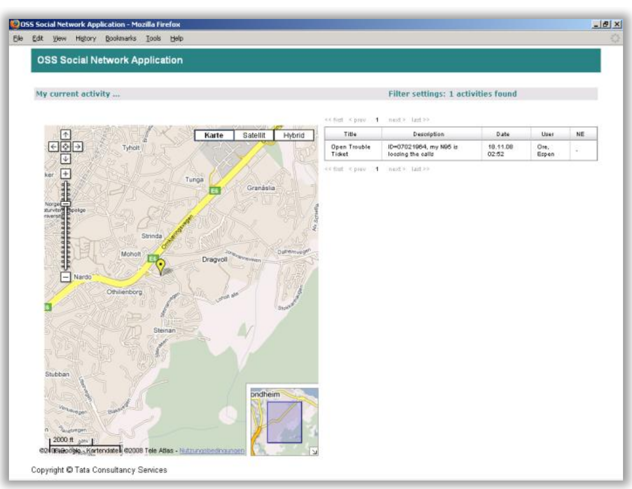


Figure 5: Activities filtered to show trouble ticket (Ex. 1)

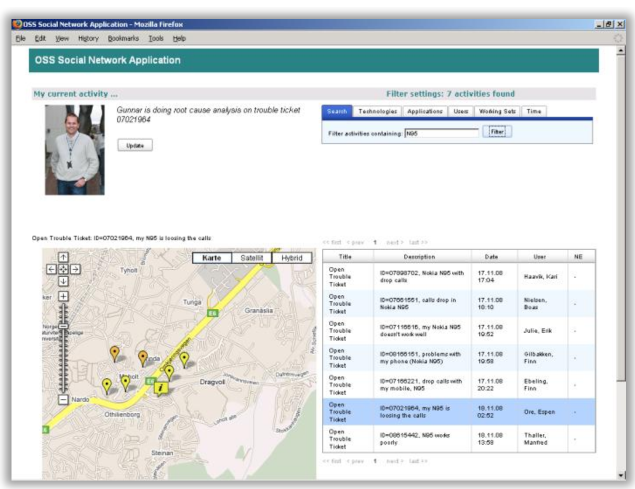


Figure 6: Other activities involving same mobile type (Ex. 1)

After having checked the performance management assurance tools of his OSS, and having found no obvious network problem in the customer's area, Gunnar turns to the Social Network application. It lists all activities of his colleagues in the past couple of days, and he hopes to find some clues here.

First, Gunnar searches for the trouble ticket number he needs to investigate. He finds an "Open trouble ticket" activity by a call center agent working in the support center. The activity is related to the location where the customer reported the problem, and is shown on the map (in our example near the student town of Moholt in Trondheim), as shown in Figure 5.

Since the performance management tools don't reveal any network problems, Gunnar's first hunch is that this might not be a network problem at all, but a fault in the cell phone's software – in our example a Nokia N95. For that reason, Gunnar searches the activity list in the Social Network application for occurrences of "N95". The search reveals a six more activities where support agents opened trouble tickets with a Nokia N95 involved. But to Gunnar's surprise, all the incidents are closely related, location- and time-wise, as shown in Figure 6. All the activities are centered around Moholt, and they all are tagged in a yellow color code, meaning they are less than two days old.

This is too much of a coincidence for a cell phone software problem. There needs to be some network problem. Since the OSS Social Network application uses the current map section as an implicit filter, Gunnar just needs to remove the "N95" filter expression. He finds 72 activities of his colleagues relating to this area, including site visits, running service tests, and acknowledgements of problems with audio conferencing (see Figure 7). The color codes indicate that the activities are spread over a period of 30 days.

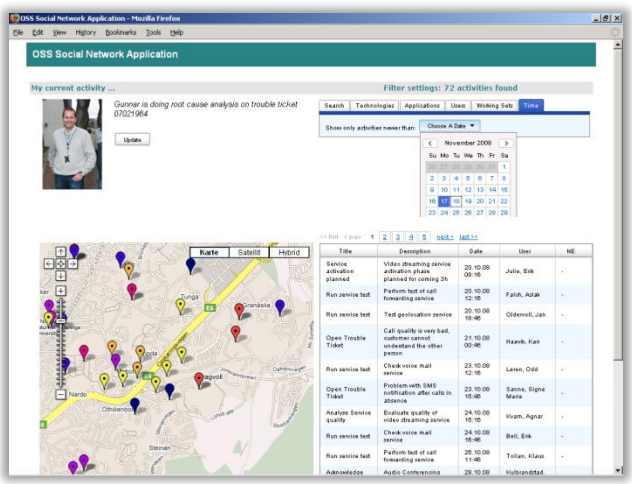


Figure 7: All activities in the identified problem area (Ex. 1)

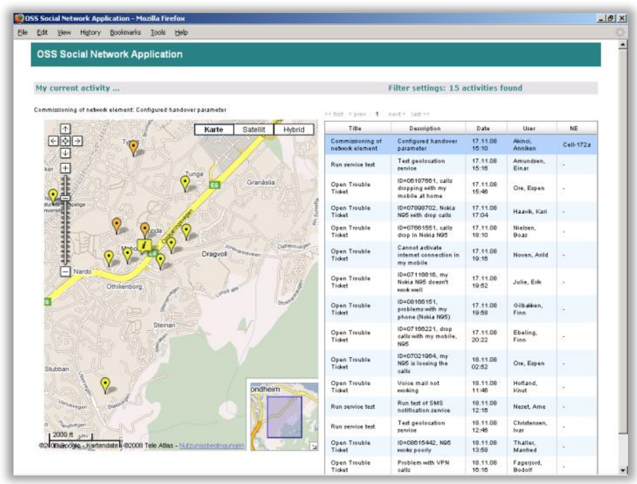


Figure 8: Likely candidate for problem cause (Ex. 1)

Gunnar is not inclined to check all 72 activities. Since he knows that the problem he is looking for must have happened in the recent past, he filters only the activities from the last three days. The remaining 13 activities are now easy to check. When reading the entry "Commissioning of network element – configured handover parameter" by Gunnar's colleague Anniken Akinci (see Figure 8), he has a hunch that this might be the cause for the handover problems with the Nokia N95. The commissioned network element is located right in the middle of the locations of complaining customers, and the first complaints start 30 min after the activity.

Example 2: Disturbed Transmission Link

Two other scenarios have been examined for the OSS Social Network application. They are briefly described here. In the first scenario, there is an obvious fault in the network: A transmission link is disturbed. This is detected immediately by the regular monitoring tools in the OSS. Transmission links are the connection between network elements positioned at various locations in the field. Often transmission links use microwave transmitters. A disturbance can have a variety of root causes that are tedious to investigate.

In our scenario, a similar data mining process as above in the OSS management activities (see Figure 9) reveal a site visit to the network element connected by the transmission link. The visit's purpose was not connected with the link; therefore the regular OSS management tools will probably not point out this coincidence. The OSS Social Network application shows the visit as time- and location-wise related to the investigated disturbance. Therefore, the investigating engineer may again get a notion that maybe the visiting field engineer involuntarily touched the microwave device during his visit, and maladjusted it.

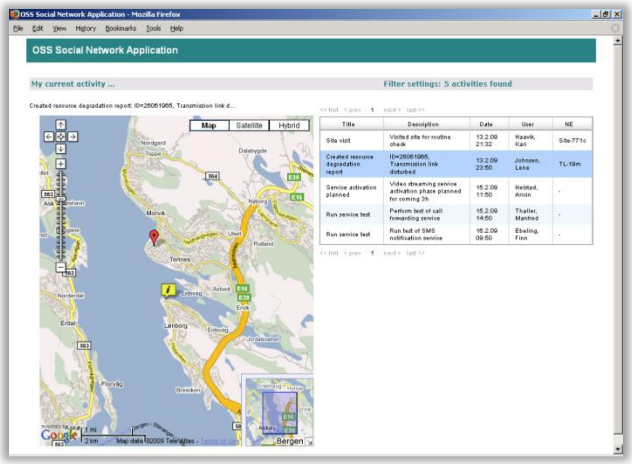
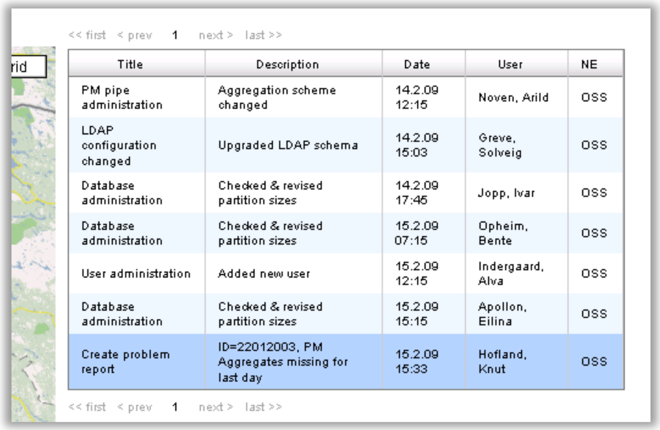


Figure 9: Likely candidate for link problems (Ex. 2)



<< first < prev 1 next > last >>

Title	Description	Date	User	NE
PM pipe administration	Aggregation scheme changed	14.2.09 12:15	Noven, Arild	OSS
LDAP configuration changed	Upgraded LDAP schema	14.2.09 15:03	Greve, Solveig	OSS
Database administration	Checked & revised partition sizes	14.2.09 17:45	Jopp, Ivar	OSS
Database administration	Checked & revised partition sizes	15.2.09 07:15	Opheim, Bente	OSS
User administration	Added new user	15.2.09 12:15	Indergaard, Alva	OSS
Database administration	Checked & revised partition sizes	15.2.09 15:15	Apollon, Eilina	OSS
Create problem report	ID=22012003, PM Aggregates missing for last day	15.2.09 15:33	Hofland, Knut	OSS

<< first < prev 1 next > last >>

Figure 10: Candidates for OSS problem cause (Ex. 3)

Example 3: Misconfiguration of the OSS Itself

In the last scenario, the problem is not related to a geographic location; therefore the map display is of no use. In addition, the possible problem causes are more ambiguous than in the previous examples. Still, the OSS Social Network application delivers valuable clues for investigation. In this case, the self-monitoring of the OSS has shown a problem with the performance management (PM) data warehouse. The aggregated data for the previous day is missing, which should have been calculated from raw data and stored to the data warehouse during the night. This is visible in the OSS Social Network application as an activity "Create problem report" (Figure 10, highlighted row).

Figure 10 shows the activity list of the application after having applied a filter on "technology/system area" (in that case the OSS itself) and time. For a support engineer familiar with the system design of the OSS PM data warehouse, at least two of the displayed activities before the problem report (i.e. of the rows above the highlighted one) are likely candidates for an involuntary system misconfiguration, possibly leading to the problem:

- "PM Pipe administration – Aggregation scheme changed" (first row in Figure 10): The aggregation scheme determines what aggregates are calculated. The change, maybe in combination with a software bug in the warehousing module, might have triggered the failure to calculate the aggregates.
- "LDAP configuration changed – Upgraded LDAP schema (second row): LDAP is in many OSS systems used to store system configuration data, also for the PM data warehouse. Even if the schema update had a completely different purpose, it might be a wise idea to check if the data warehouse configuration is still intact. The failure to load configuration parameters might have lead to the problem in aggregates calculation.

DISCUSSION

Strengths

As the thought experiment has shown, an Enterprise Social Network Platform can be an efficient tool for increasing informational transparency within a large organization, and is especially suited for supporting troubleshooting situations. Its filter is efficient because of its simplicity. It just pre-filters the information, and allows data mining operations. Therefore, it builds upon the experience and intuition of the operational personnel, instead of a sophisticated rule engine. This makes the system flexible and cost-efficient.

Other cost benefits are the cheap hardware (the application is not mission-critical, and doesn't require expensive failover mechanisms) and the commodity software components used. Integration costs into an existing IT infrastructure are presumed to be low, since standard interfaces are used for integration.

Challenges and Open Issues

On the challenges side, the question about the limits of unobtrusive integration remains to be answered by a practical trial. For other technical areas than telecommunications and OSS, proper standards need to be identified that can act as integration interfaces. Not all technical domains have standards like eTOM in place.

Beside the technical challenges, social issues play an important role when introducing an Enterprise 2.0. This includes cultural and legal challenges, both of which often reflect typical social aspects.

Wikis or other Web 2.0 tools are usually used on a voluntary basis. This principle strictly needs to be maintained with the platform proposed in this paper. Otherwise, there will be several social and even legal issues. Legal challenges are given regarding the aspects of data protection and data privacy. Data protection usually is connected with prevention of diffusion of company insights to outsiders, and prevention of misusage of internal systems by others than beneficial owners. If a company would attempt to introduce such tool on a mandatory base, and the user has no influence on the individual-related data to be stored within the enterprise social network, the company would face severe resistance from works council organizations, or even legal battles in court.

But even if participation in such a system is voluntary, there might be a social pressure on individuals, as all activities are logged automatically within the system. Global usage of such a platform leads to an extensive increase of transparency of processes and the involved participants' actions. Therefore, a feeling of being watched might occur for those who participate. If there is no intrinsic motivation and willingness to cooperate by the users, this might lead to low acceptance of the platform, or even a disturbance of the general work climate.

A possible way out of this dilemma is to allow the user to configure which kind of activities can be logged by the social network platform. In turn, he receives notification only of those activities of others that he has approved of for himself.

The adoption and usage of social software is based on the principle of mutuality. Collaborators that do not trust the system, or don't have any experience with social software, might be lacking incentives to integrate into the network. Specific cultural challenges might occur if the threat of losing face or being blamed for a mistake is heavily founded in a society. Therefore it is necessary that users are not only focused on the individual benefit, but are willing to participate to strengthen the enterprise.

On the other hand, positive effects like the ability to be identified as an expert in a specific field should be mentioned as well. Utilizing the social network in such a context might lead to user contributions, because users expect to be helped by other users when necessary in the future. Besides, to be regarded as a process owner might increase motivation of users to contribute (Koch, Richter 2007).

Outlook

The following aspects can be considered as an outlook for further developing the idea proposed in this paper.

Productified Version Should Come as Full-Fledged Social Network Platform

The enterprise social network platform proposed in this paper has been somewhat "stripped down" to the essential elements to demonstrate the new aspect of it as a tool in system management. However, when such a platform would indeed be productified, it would make sense to roll out a full-fledged social network, increasing the value for the users, and therefore also their buy-in to the concept. Social platforms for companies in general are expected to show the following functionality (McAfee 2006; Hinchcliffe 2007; Hippner 2006):

- User-focused: assists the users in their activities
- Search: allows users to search for other users or content
- Links: groups similar users or content together
- Signals: allows people to subscribe to users or content via RSS feeds
- Self-organization: based on self-organization of tasks

- Network-oriented: all content must be Web-addressable
- Social: stresses transparency, diversity and openness

So, a good tool set available in such a platform should include the following social software tools that have been adapted for business purpose (Koch, Richter 2007; Back, Gronau, Tochtermann 2008):

- Hypertext and unstructured search tools
- Wikis to support knowledge management processes
- Weblogs and podcasts for storytelling and information distribution
- Social bookmarking for tagging and building organizational folksonomies
- RSS for signaling
- Collaborative planning software for peer-based project planning and management
- Prediction markets for forecasting and identifying risks

Direct Integration into the Management Applications

Instead of displaying the user activity notifications in a dedicated social network application platform, the user experience could be improved by integrating the activity list directly into the system management application itself. In the given example of an OSS for mobile networks, it would make sense that a user of a network view tool can select a node representing a network element, and open the list of activities related to this element. This would integrate the social network functionality seamlessly into the user's normal workflow.

Such a direct integration is difficult to perform in an unobtrusive way. Either it is implemented on a case-by-case base for selected system management tools, or techniques like using browser plug-in are used.

Inclusion of General-Purpose and Custom-Built Applications

Adding an open API to allow administrators or even users to deploy own, company-specific applications to the platform would increase the versatility of the approach. Taking again the OSS example, adding applications that post information on regional weather or news events into the system would add value in certain situations.

A network element disturbance may be easy to explain for a monitoring engineer if he or she knows that there has been a local hailstorm nearby. Or, to construct a slightly more far-fetched example, a series of customer complaints about dropping calls are easily explained if the engineer learns about a bomb alarm in the vicinity, where the police used a jammer to disarm possible detonators, and where this jammer distorts all mobile communication in the area.

SUMMARY AND CONCLUSION

As shown in this paper, a customized enterprise social network platform can increase the management efficiency for complex and distributed technical systems. The concept can be applied regardless of the nature of the technical system – let it be a communication network, a manufacturing facility, or an enterprise IT landscape. The approach has manifold benefits, but also some open issues, which need to be further researched.

The enterprise world shows a strong trend towards providing a "general purpose" social network platform to the employees, to increase communication and inter-personal networking. An implementation of a platform as proposed in this paper, combined with the "traditional" elements of social network platforms, would effectively serve both purposes.

REFERENCES

1. AIIM (2008) What is Enterprise 2.0? http://www.aiim.org/What-is-Enterprise-2.0-E2.0.aspx [2009-02-19].
2. Back, A., Gronau, N. and Tochtermann, K. (2008) Web 2.0 in der Unternehmenspraxis, Oldenbourg, Munich, Germany.
3. Bauer, N. and Mandl, P. (2007) Agiles Informationsmanagement, *HMD - Praxis der Wirtschaftsinformatik*, 255, 88-96.
4. Bente, S. and Hegle, G. (2008) Using Web 2.0 social network APIs in OSS applications, *Telemanagement World Americas Conference*, November 18-20, Orlando, FL, USA.

5. Berners-Lee, T. (2006) developerWorks Interviews: Tim Berners-Lee. http://www-128.ibm.com/developerworks/podcast/dwi/cm-int082206.txt [2009-02-19].
6. China Mobile (2008) Operation Data - Key Operational Statistics. http://www.chinamobileltd.com/ir.php?menu=12 [2009-02-20]
7. Eslambolchi, H. (2008) Power of technology to transform the future, *Keynote of Telemanagement World Americas Conference*, November 18-20, Orlando, FL, USA.
8. GSMA (2008) Mobile Infolink - Market Data. http://www.gsmamobileinfolink.com/market_data.aspx [2008-02-20]
9. Hinchcliffe, D. (2007) The state of Enterprise 2.0. http://blogs.zdnet.com/Hinchcliffe/?p=143 [2009-02-19]
10. Hippner, H. (2006) Bedeutung, Anwendungen und Einsatzpotenziale von Social Software, *HMD - Praxis der Wirtschaftsinformatik*, 252, 6-16.
11. Karla, J. (2007) Geschäftsmodelle und technologische Realisierung von Web 2.0-Publikumsdiensten, *Information Management & Consulting*, 1, 30-34.
12. Kaul, K. (2005) Web 2.0 - Phantom oder Phänomen? http://www.dw-world.de/dw/article/0,2144,1790308,00.html [2009-02-19].
13. Koch, M. and Richter, A. (2007) Enterprise 2.0, Oldenbourg, Munich, Germany.
14. Koplowitz, R. and Young, G. (2007) Web 2.0 social computing dresses up for business, Forrester Report.
15. McAfee, A. (2006) Enterprise 2.0: The Dawn of Emergent Collaboration, *MITSloan Management Review*, 3, 21-28.
16. O'Reilly, T. (2005) What is Web 2.0? http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html [2009-02-19].
17. TM Forum (2008) Business process framework (eTOM) for the information and communication service industry, Release 7.5, GB921.
18. Tredinnick, L. (2006) Web 2.0 and Business: A pointer to the intranets of the future? *Business Information Review*, 23, 228-234.
19. Young, G. (2008) Top Enterprise Web 2.0 predictions for 2008, Forrester Report.