Smart Media: Bridging Interactions and Services for the Smart Internet

Margaret-Anne Storey, Lars Grammel, and Christoph Treude[*]

University of Victoria, Department of Computer Science
`{mstorey,lgrammel,ctreude}@uvic.ca`

Abstract. This chapter describes a need for Smart Media to enhance the vision of the Smart Internet. Smart Media is introduced as a mechanism to bridge Smart Services and Smart Interactions. Smart Media extends the existing notions of Media in HCI such as Hypermedia, New Media, Adaptive Hypermedia, and Social Media. There are three main contributions from this paper: (1) A historical perspective of media in HCI and how media could benefit from smartness; (2) through some high level sample scenarios, a proposal for Smart Media to meet the vision of the Smart Internet; and (3) a detailed example of how Smart Media could play a role in software development. The paper concludes by outlining some of the key challenges that need to be faced in realizing and applying Smart Media objects.

Keywords: Smart internet, smart interactions, smart services, smart media, social media, web 2.0, intelligent user interfaces, computer supported cooperative work, human computer interaction.

1 Introduction

The vision for a smart internet, according to Ng, Chignell and Cordy [1], arises from a shift away from a server-centric model to a user-centric model. It realizes that the user's model is paramount in terms of supporting the users' interactions, and consequently a new kind of session is needed that centers on the user's perspective and context [1]. That is, a user's interaction on the web should be situated within a socio-technical perspective that involves multiple web services, context about the user, as well as collaborations with other people. Such elaborate interactions thus require a "dynamic social binding" [1] of web interactions with web applications that supports collaboration across individuals, as well as knowledge about the user's needs.

Ng *et al.*'s vision of the Smart Internet requires both "Smart Interactions" and "Smart Services". Smart Interactions focuses on developing an advanced user model on the web that captures the users' contexts and their matters of concerns. Smart Services refers to the algorithms that will orchestrate the web as a cohesive platform thus enabling the advanced user models required for smart interactions [1].

---

[*] The authors are supported by grants from the Canadian Institutes for Health Research (CIHR).

In this chapter, we explore Smart Media as a potential third component of the Smart Internet. Media, the plural of medium, sometimes refers to channels or systems of communication, information, or entertainment[1,2]. The term is also often used to refer to artifacts that are communicated along a given channel, such as a specific video clip or newspaper article.

The need for smart media on the smart internet is based on the observation that users' interactions with web services usually involve information artifacts that play a central role in the current, but also future, interactions of the users with the web. For example, when booking a flight over the web, the artifact created is an airline booking. This booking can be transmitted along various channels, such as by email or by posting on a secure website. The email and website channels are media channels that are used to mediate the users' interactions with the services, as well as mediate their interactions with other stakeholders in the booking.

In this chapter, we explore if there are potential opportunities for and potential advantages of adding "smartness" to the information artifacts themselves. Rather than referring to them as just information artifacts, we suggest referring to these enhanced information artifacts as smart media objects. Essentially, we propose that smart media objects can become a way to bridge smart interactions and smart services by reifying and aggregating the user's context with information resources and services that will satisfy the user's needs.

In order to explore the proposed concept of smart media, we look back at the history of different forms of media in Human Computer Interaction (Section 2). We then consider what the requirements may be for media that is "smart" in Section 3 and refer to some potential applications of smart media. In Section 4, we explore how smart media can be realized in the architecture of the Smart Internet. In Section 5, we present detailed scenarios of how "Smart Media" could play a role in collaborative software engineering and how developers may benefit from smart media objects. Finally, in Section 6 we articulate the many challenges that must be overcome to design smart media for the Smart Internet and outline future work. We conclude the chapter in Section 7.

2   A Brief History of the Role of Media in Human Computer Interaction

In this section, we explore the historical perspectives of media, hypermedia and the more recent phenomenon of social media.

2.1   From Media to Hypermedia

The term "Media" is often used to refer to different but related concepts. For example, media may refer to channels or systems of communication, information, or entertainment. "Mass media" more specifically refers to media that is communicated to a

---

[1] As opposed to Mass media, communication (as newspapers, radio, or television) that is designed to reach the mass of the people (http://www.merriam-webster.com/dictionary/mass+medium)

[2] Definition of medium: http://www.merriam-webster.com/dictionary/medium

wide population. Examples of mass media include newspapers, radio and television. In Human Computer Interaction systems, examples of media artifacts include text files, photographs and emails. The term "new media" is often used to refer to the emergence of digital media and networked information resources that occurred over the past 30-40 years. New media are digital artifacts such as videos that can be interactively manipulated and linked over the internet.

We note that the use of the term "Media" often blurs the distinction between **channel** (e.g. email system, youtube), **data artifacts** (a specific email or newspaper article) and the **representation** that is used to communicate the data (e.g. video, text, graphics). More succinctly, a channel is a mechanism whereby a given representation of a data artifact can be transmitted to the user. The selection of the appropriate channels and the manner in which the artifact is represented is critical in terms of meeting the user's needs. We note that it is not always possible to distinguish representation from channel, e.g. in the case of a Twitter feed, the representation is fixed (text of length 140 characters) although different tools could present the feeds in a different way (e.g. by expanding pictures or showing clips of embedded links in the feed). We provide the following high level definition of media to guide our future discussion:

Media = Artifact + representation + channel

A major challenge in HCI research has been to find suitable tools and environments for creating, searching, manipulating, consuming, annotating, sharing and analyzing digital media artifacts. As early as 1945, Vannevar Bush noted that *"publication has extended far beyond our present ability to make real use of the record"* [2]. Indeed he proposed a system called the Memex, an external memory device where individuals could store and index all their personal books, records and communications. Users could construct "trails" through the material, and annotate their resources with margin notes, comments, keywords and cross references to other material. Although the Memex concept was based on Microfilm, many saw it as a precursor to "hypertext".

The Hypertext and Hypermedia terms were later coined by Ted Nelson in 1965 [3]. Hypertext refers to a non-sequential collection of text [4]. It can be modeled as a directed graph where each node represents a certain amount of text. Hyperlinks (directed arcs) connect the nodes in the graph. Hypermedia extends the term hypertext to include graphics, audio, video, as well as text and hyperlinks, to create a "non-linear medium of information" [5]. The term multimedia is a broader concept that includes hypermedia, as well as non-interactive mixed media.

Although never fully implemented nor adopted, Nelson proposed the Xanadu system. Xanadu was to be a system that could store all the world's literature, with the ability to link from any substring to another, and where the most recent version would be retrieved (but older versions would be retained and would never be deleted). Other concepts in Xanadu included author attribution and transclusion, the inclusion of a part of a document into another document by reference. Although not fully implemented, Nelson's concepts of Hypertext and Hypermedia are fundamental concepts to all web-based information systems today. However, Nelson's disappointment in some of the limitations from today's web is evident in this quote of his: *"The 'Browser' is an extremely silly concept-- a window for looking sequentially at a large*

parallel structure. It does not show this structure in a useful way." Our observations of how users interact on the web have indicated to us that the browser is also a poor mechanism for user interaction with media artifacts that supports a user's task that crosscuts multiple sessions on the web. Part of the issue is that the user further wants to use different channels for viewing and interacting with their media artifacts, but there is currently very poor integration or awareness across these channels[3].

Despite some drawbacks with today's web, it is of course highly successful and widely used. Some of the reasons cited for its success are that it is built around an open systems approach with standardized specifications for interoperability. It is also backwards compatible (for legacy data) and control over content on the web has been decentralized [6]. Today's media artifacts are typically embedded in a hypermedia system that consists of a web of links to other media objects.

The more recent development of Web 2.0 technologies and a change in philosophy have led to a new form of media, that of Social Media.

2.2   From Hypermedia to Social Media

The emergence of Web 2.0 technologies and a change in philosophy in how users participate as authors as well as consumers over the web, have resulted in a paradigm shift in how users (and their social networks) create, gather, annotate, share and analyze information [7, 8, 9]. This new model has led to the emergence of "Social Media". Social media is often referred to as the many-to-many broadcast mechanism, and the implication of its creation along with the usage of Web 2.0 technologies are widespread, but as of yet quite poorly understood.

Web 2.0 or social software as it is often referred to, is built around an "architecture of participation" that aggregates both explicit user contributions (e.g. created artifacts) and implicit user contributions (e.g. navigation trails). Web 2.0 implies that processes and tools are socially open, and that content can be used in several different contexts. Web 2.0 tools and technologies support interactive information sharing, data interoperability and user centered design. Web 2.0 affords the creation of social media, where content (information artifacts) can be read, written, and transformed into other content by a many-to-many collaboration if desired. One of the most important implications of Web 2.0 is a significantly enhanced user experience, as well as user control over the applications that they use and the data they create.

Example social media channels include wikis, blogs, mashups and feeds, where specific artifacts are wiki articles, user blogs and instances of mashed up services. The media channels are being used by users and communities of users to help organize, manage and categorize artifacts in an informal and collaborative way.

Referring back to the airline ticket example: how could social media improve the interactions and the services underlying the ticket? Some suggestions are that it may be desirable to use social media to assist in creating the airline booking, by helping the user choose which airline or route to select during the booking (note: this is already supported by websites such as expedia.ca). But social media could also be used to enhance the experience of the booking once underway by linking in social media that informs the user as they experience their trip (e.g. reviews recently uploaded to

---

[3] The recent introduction of the OpenGraph protocol by Facebook and their collaborators is addressing that very issue, see http://developers.facebook.com/docs/opengraph.

tripadvisor.com or just in time transportation issues discussed on Twitter). Moreover, the user could use the booking itself as a portal to add to this user generated content if desired, and thus help other users with their travel plans in real time. Some of these enhancements to ticket bookings already exist – for example, expedia.ca ties in many of these services so that updates to the itinerary can be made in real time. However, such enhancements are made following a server centric point of view with only limited context about the user and the user's task playing a role in how new information and services are pushed to the user. This is where more "smartness" is required to put the user at center of the booking artifact, and thus greatly improve the user experience.

3   Towards Smart Media for the Smart Internet

As mentioned previously, we propose that the smart internet needs not only smart interactions and smart services, but also the notion of smart media. But what do we mean by "smart"? Here we can draw inspiration from the research community of "Intelligent User Interfaces". According to Maybury [10], *"Intelligent user interfaces (IUIs) are human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse and media (e.g., graphics, natural language, gesture)."* Many of the goals articulated for IUIs match those of the smart internet, thus we can learn from and build on research from that community.

   "Smartness" in user interfaces these days is appearing in many forms. For example, the use of recommendations [11], adaptive interfaces [12], and the semantic web [13], are just a few areas where "intelligent" support for the user is being explored. While all these efforts are making large strides to improve the user experience, what is perhaps lacking is a cohesive plan of how interactions, media and services can together improve how the internet fulfills users' personal and social needs. Users still suffer from the lack of integration of services, difficulties in tracking tasks (especially across services) and challenges personalizing web pages to suit their needs [1]. Many of these concerns stem from insufficient knowledge about users, their social network and the tasks they need to complete.

   The lack of a consistent user model across services is also a significant reason for difficulties and barriers to an effective user experience. One research area, that could be helpful in this regard, is the area of User Modeling (see http://www.um.org/), this research topic explores how to develop cognitive models of human users describing their knowledge and skills. One example of user modeling underlies "adaptive hypermedia" – hypermedia that adapts to meet the user's goals, abilities, interests and knowledge [14]. Thus, User Modeling is another community that we should look to for inspiration in developing the Smart Internet and in particular for supporting Smart Interactions. Indeed, capturing, updating and leveraging the User's Model is a crucial aspect of the proposed Smart Interactions on the web.

   But how can the media artifacts and media channels offer aspects of "smartness"?

   We already see examples of "smartness" emerging in media channels. Consider the youtube channel where we benefit from recommendation systems, and in email we benefit from spam detection that is improved through collaborative filters. As another example, the use of Twitter during the earthquake crisis in Haiti [15] demonstrates

how the Twitter media channel can be used to harness collective intelligence in helping rescue efforts. Such examples can be further augmented through the use of agents and services, as well as information about the User's model (such as history, user preferences etc) to improve the user experience.

What we propose in this paper is to also push smartness into the media artifact itself.

Let's consider the flight booking example again. When I book my flight, I can get a copy of the booking by email, which is then added to my calendar and I can access a copy of it on a secure website. There are two or more copies (representations) of the booking. The booking as presented on the website channel is likely to be updated if changes are made, but the email version and the calendar versions will not be updated. This is one problem that can cause significant difficulties for users as they have to manage multiple versions of the booking and remember which one is valid and furthermore may miss important updates if a non-live version is used. This is not an unusual scenario, many media artifacts are often constrained to be used across just one channel, and can't easily cross the boundary between channels, and only some channels will support live updates.

Another issue is that although the website version of the booking can customize the booking according to the user's preferences stored on the website, the website channel may not be privy to information generated from other services that may impact the user's travel plans (for example, imagine that the user is waitlisted for surgery through another service, and the user does not wish to carry through with their travel plans if it conflicts with a surgical opening that arises).

Rather than requiring all services to know about all other services that may be relevant to a user, the currently proposed model of the smart internet assumes there is a user model that the services access. The services thus update the artifacts as needed according to the user's model which should ensure that consistency is achieved across all the services and media artifacts relevant to the user.

What we propose is that some of the smartness required should be pushed into the media artifact itself, and that the media artifact should play an intermediary role between the User's Model (smart interactions) and the services they use. If some of the "smartness" is pushed down into the artifact, the artifact that is created by, for or with the user can adapt to and transform a user's needs. This is achieved by having a model of how the artifact should behave. The artifact behavior is made possible through the use of services and by interacting with the User's model.

Rather than the term "artifact", a better term may now be *smart media object* – to emphasize that it represents data, representation options, channels and intelligent services. The smartness comes from the fact that a smart media object has a model that describes how it should behave to meet the users' needs, it can access information about the User's Model and leverage services that matter to the user. From the user's point of view, there is just one booking artifact that can be displayed on different channels (using different representation techniques if desired), and it can be updated through a variety of means so that *just-in-time* information that matters to the user will be available. Such an approach would reduce friction in how the user interacts with the internet but would also greatly enhance the user's experience through additional service integration.

We now revisit our high level definition of media from above, and expand on it to provide a high level definition of smart media object as follows:

Smart media object = Data + representation + channel + *intelligence*

The following provides natural language definitions of the terms *smart media object* and *smart media channel:*

> **Smart media objects** *are socio-technical artifacts that are aware of user(s) context(s), they aggregate data from one or more services and through the use of services and through dependencies with other smart media objects can anticipate, participate in, adapt to, react to, evolve and transform user(s) needs.* **Smart media channels** *are used to transmit data between the user's devices and the smart media objects that are linked to the user.*

We expect these definitions to evolve as we explore these concepts, but for now they at least capture the intent underlying the proposed benefits of smart media to support the Smart Internet.

The concept of smart media is of course not entirely new; adaptive hypermedia that adapts to a student's learning needs exhibits some characteristics of "smart" media that we propose and there are very likely other examples of this concept. What we propose is that a unified and widely adopted concept of a smart media object can support an improved mediation of the user's interactions with smart services. We emphasize that the smart media objects should not be tightly coupled to specific services nor dependent on specific channels, but according to the user's needs can be accessible anyhow, dynamically updated and be live on the channel that is most appropriate for the user's task and context.

Although this proposal may seem a utopia in HCI, there are many current research efforts that lead us in this direction, such as Web 2.0, the semantic web, service oriented architectures, cloud computing and advanced user modeling techniques. In the next section of this chapter, we present a potential architecture that fits with this vision.

4   A Proposed Architecture for the Smart Internet That Supports Smart Media

So far we have described what we believe could be the benefits from having smart media objects for improving the user experience on the web, but not discussed much how these smart media objects would interface with smart services and smart interactions. In Figure 1, we first show a high level architecture of the proposed Smart Internet.

In Figure 2, we add to this architecture and propose how Smart Media Objects could fit into this architecture. Smart interactions are implemented using the notion of a personal agent bound to a model of the user. There is one such agent and model for each user; the model is updated using information sent to it through media channels that the user opts to use, through service updates and through sensor inputs to the model. The agent will use the user's model to decide which data captured by the smart media objects should be communicated to the user, and which media channels should be used for communicating that information.

To relate our previous example of a flight booking to this architecture, we describe how it could be instantiated using the model in Figure 2. The travelers booking their
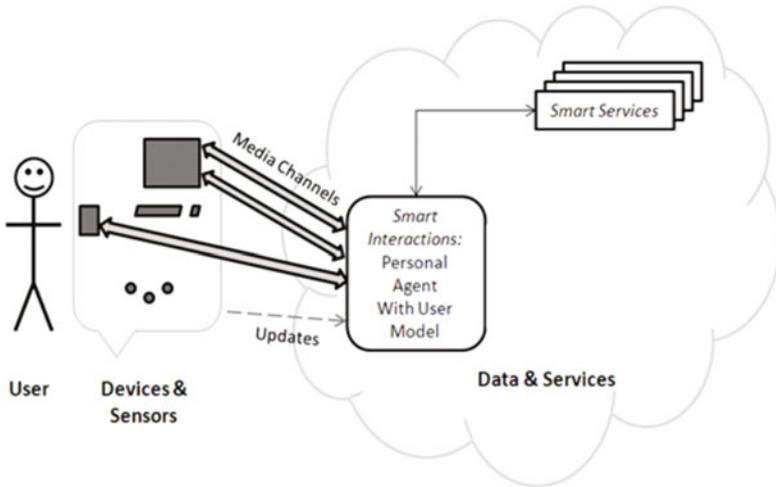
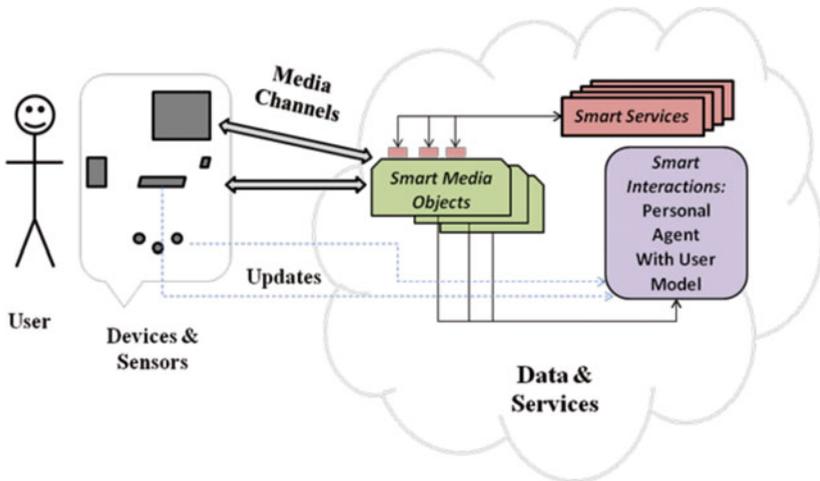**Fig. 1.** Architecture for the Smart Internet.

Fig. 2. Smart Internet Architecture with Smart Media Objects.

travel are instances of the user. Users have a smartphone (with GPS services) and a laptop as devices at their disposal. They have previously booked their flight using expedia.ca and their flight booking is now represented as a smart media object. The user's model captures the preferences of a particular user, including the fact that the user prefers an aisle seat whenever possible. The flight booking object notes that the user is currently assigned to a window seat and using one of the smart services associated with the booking object, periodically polls to see if an aisle seat comes available. When an aisle seat is available, the flight booking data is updated to reveal a new seat has been assigned. The booking object refers to the User's model and notes that the user wants to be informed when a change is made to the seat assignment using an instant message channel connected to the User's smartphone.

This last example demonstrates how the media object *pushes* information to the user and makes changes to the artifact (the flight booking) to benefit the user. In order to "know" which changes should be made, the media object has an internal model keeping track of the state of the user (that is relevant to the artifact), the state of the booking itself and references to services that it may pull information from, or registered services that can push information to the media object. The smart media object's model is not represented in Figure 2, but the services it uses for both pushing and pulling information from the services are shown as connectors to the Smart Media Object.

In order to demonstrate more aspects of the architecture, we present a second scenario based on user-initiated *pull* operations. Assume the user is at the airport and has not checked in her baggage yet. The user initiates looking at the booking through her smartphone. The personal agent passes the context (smartphone, location at airport, time) into the view operation of the booking (which knows the user has not checked in her luggage). The booking media object returns a representation of itself that emphasizes the location and name of the check-in counter. The personal agent indicates to the booking that the user's web browser channel should be utilized to render a route from the user's current location to the check-in counter with additional information about the booking. After the user has checked in, but is not through security, the same operation would result in a map with the next security gate location, time until boarding and security instructions as well as the current average waiting time at the security gate. After the user is through security (user location has now changed), the booking would send a representation of itself also through the web browser channel that emphasizes the gate location and boarding time.

The examples just presented are to illustrate how the smart media objects can play a role in the architecture of the Smart Internet. In the next section of this chapter, we look at a specific domain, that of collaborative software development, and further explore how smart media objects could play a role in that domain.

5   Smart Media Objects for Software Development

In this section we propose that software development could be an appropriate test environment to explore the concept of smart media objects. First of all, software developers tend to be early adopters of new technologies[4]. Indeed, social media technologies are causing a paradigm shift in how software is developed. Developers routinely blog [16], use twitter [17], use social networking [18] and use social coding sites such as github.org and heroku.com[5]. When searching for information they make use of Google search and websites such as www.freshmeat.org and www.stackoverflow.com.

Secondly, modern IDEs and IDE features under research development already benefit from "smartness". The notion of "content assist" is now routinely used in search engines and development environments. Recommendation engines for code samples, developer expertise, related bugs (work items) and test bases are currently an

---

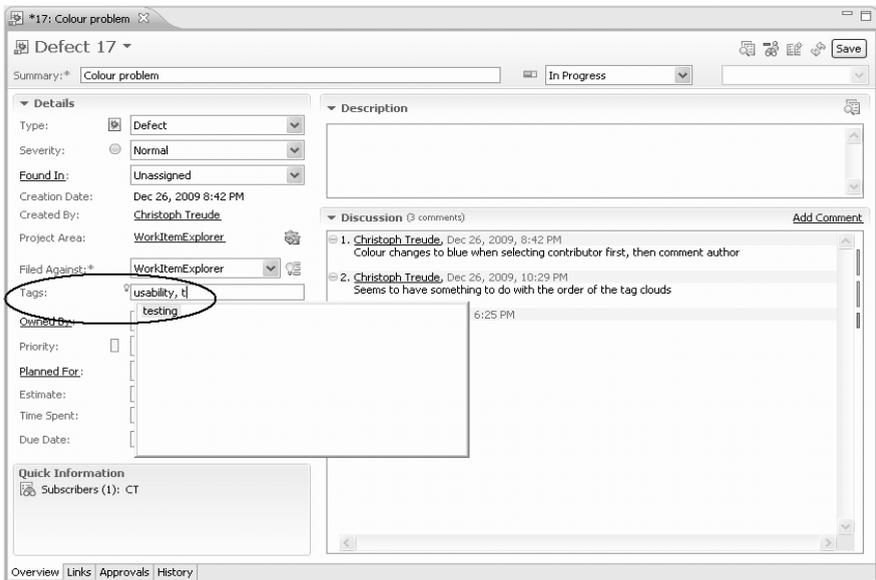[4] This is particularly so for younger developers.
[5] See http://sites.google.com/site/web2se/

active area for research in software engineering[6]. Along with this, there is active research on how to improve search during development[7]. Recent research on IDE improvements investigates how more knowledge of the user and their tasks can be used to improve recommendation engines [19] and how such information could lead to an improved knowledge based IDE [20]. Chapter Four in this book further explores how smart interactions can improve interruption management in software development [21].

By partnering with groups of developers to try out the ideas underlying smart media objects, these developers can participate in helping us determine the requirements for smart media objects and furthermore provide feedback to us on the concepts underlying the smart internet. We propose that three social media artifacts from software development, work items, dashboards and feeds, can be transformed into smart media objects that will improve collaborative development activities. We consider these artifacts within the context of the Jazz development environment (an IBM product).

5.1 "Smarter" Work Items

Modern software development environments typically have explicit tool support for managing tasks. For example, Jazz[8], a collaborative IDE developed by IBM, has tool support for managing "work items", where a work item is a generalized notion of a development task (see Fig. 3). Work items are assigned to developers, are classified using predefined categories, and may be associated with other work items. Jazz work

Fig. 3. Work items in Jazz, the circle shows how developers can freely tag work items.

---

[6] See http://sites.google.com/site/rsseresearch/rsse2010
[7] See http://scg.unibe.ch/wiki/events/suite2010
[8] www.jazz.net

items also have some web 2.0 tool support to address social aspects. Specifically, Jazz supports a discussion thread and a lightweight "tagging" mechanism. Using this latter feature, developers can freely associate user defined keywords with work items. In our previous work [22], we showed how the tagging feature of work items has been eagerly adopted by developers.

Work items can be viewed in the client application of Jazz or through a web browser. Developers can set up feeds on work item events as well as see views on the work items in dashboards. Dashboards and feeds are discussed further below.

Some of the challenges facing developers that use work items include the following:

- *Notifications:* Developers have to customize how they should receive notifications of events related to specific work items. This can be quite cumbersome to do and there is a tension between having too much information (overload) and missing out on important events that may be important to them. This latter issue relates to developers maintaining appropriate awareness during collaborative development activities. McGrenere and Li's chapter in this book investigates the related issue of interruption management [21].

- *Finding out about related work items:* There are many different possible relation types possible between work items (such as child of, depends on, and related to). The Jazz work item interface provides support for navigating related work items – but the developer has to know about the related work items to encourage this exploration. Moreover, many work items may be interrelated but no explicit relationship has been created between them. Thus support for finding related work items automatically would be a useful improvement to the tool support. The current research on recommending bugs shows promise for practical application.

- *Expertise location:* A single developer within a large software project can't possibly have expertise on all aspects of the project. As work items are created by a developer, she should be able to quickly find team members with the required expertise. Currently the developer has to manually search through the logs or ask people to find this out. There is current research on expertise location that could be used to provide this information.

- *Managing order of work, articulation of work:* Although work items are already an important cog in the wheel of team based development activities, they could be further enriched if they actively played a role in the overall work plan. Jazz currently has some support for this through a process management component, but again much of this has to be manually handled by the developer but aspects of this could be automated.

Some of the problems described above could be eliminated or at least reduced through smarter work items. But what would a smarter work item object look like? We use our definition of smart media object above to describe the architecture of a smart work item object:

Data: A work item artifact has a predefined structure with a set of fields as follows: summary (free text), priority (levels predefined), severity (from a given set of choices), component name, tags (freely assigned), comments (free text, shown in sequence), owner, creator, subscribers, state, team area, and a description (text). It also has a set of typed links to other work items and a history of how it has changed.

Channels: The work item can be displayed to the user using the Jazz client channel, or through the Web browser channel. Alternatively a feed channel may be used to communicate events on the work item.

Representations: What is represented and how it is represented depends on the channel but the representation should also be adapted to fit the user's needs. Text and graphics are used primarily in the web and client versions of the work item representation, but just text representations of some of the elements in the work item is used in a feed.

Intelligence: What we propose is that the smart work item media object should have associated with it a number of heuristics (i.e. a model) of how it should be updated and how it should send information to the user. These heuristics should work in concert with the user's model (smart interactions) as well as respond to events from services and make explicit calls to services as needed.

The above description of a smart work item object could play a role in the management of collaborative work as follows: when a work item is closed, a developer that may have related work to do should be notified. A feed can inform the developer that the work item status has changed, or alternatively, if the developer is currently reading emails, he may receive an email providing some more information on the work item that was closed. The work item should then keep track of which information has already been communicated to the developer. The work item object may further let other developers know who has been informed. The behaviours that the work item exhibits will depend on the heuristics associated with it, its own internal model of the work item model, the user's model and the available services that are associated with the work item object.

5.2 "Smarter" Dashboards

Dashboards are information resources that support distributed cognition; they are crucial to many business intelligence applications. Dashboards are also used in software development [23]. For example in the Jazz IDE, dashboards are displayed and configured using the web interface. They are intended to provide information at a glance and to allow easy navigation to more complete information (such as to a work item). By default, each project and each team within a project has their own dashboard; and an individual dashboard is created for each developer when they first open their web interface. Figure 4 shows an example dashboard.

A dashboard consists of several viewlets. Viewlets are rectangular widgets displaying information about some aspect of a project. Each viewlet is an instance of a viewlet type. The actual content shown in a viewlet depends on the viewlet type, e.g., visual representations of the current workload or a list of members on a team, as well as the way the particular instance has been configured. Developers can add viewlets to their dashboards and configure the viewlets using different parameters. Viewlets can be organized into different tabs within a single dashboard.

By default, dashboards only display general purpose viewlets containing information about developers and teams, and links to general feeds. In addition to individual customizations of dashboards, project managers or component leads can customize the default settings. Normally the development manager of a project is in charge of

Fig. 4. Dashboards in Jazz.

updating the project dashboard, the component leads are responsible for the team dashboards, and individual developers change their own dashboards.

With dashboards, different people want to see different things depending on their role, development tasks to be supported, where the team is in an iteration (e.g. just before a release developers need to know about the must fix items), and the size of the team. We conducted a study of how dashboards are used in a collaborative development project [23], and one of the challenges with using dashboards is that users have to customize them according to their needs (as described above). Furthermore, as the project progresses, they need to further customize the dashboard as their information needs change.

Another issue is that in Jazz there is another feature called Team central views (shown within the client) that overlaps much of the information shown in the dashboard within the web browser. Ideally, these two views should be representations of the same data and once one of them is customized for the user, the other one should be customized in similar manner. Moreover, if the user has viewed information in one of these views, the other view should reflect that the information is no longer new to that user.

These issues of customization and managing state about what the user has already viewed could be solved by adding "smartness" to the dashboard. We again use the high level definition of a smart media object to help propose what a smart dashboard media object may look like:

Data: The information to be displayed by the dashboard would include data from work items, build results, team membership and development events.

Channels: In this case the relevant channels for the developer may be a view in the client IDE application, a view in a web browser, or a special view designed for a mobile device.

Representations: The representations used in the dashboard will depend both on the needs of the user, but also on the data to be represented. Currently dashboards make use of charts, tables, lists and tag clouds.

Intelligence: For smarter dashboards, we again propose that this smart media object should have associated with it a number of heuristics for how it should be updated and how it should send information to the user and a model of the dashboard object that captures its state. The dashboard object heuristics should work in concert with the user's model (smart interactions) as well as respond to events from services and make calls to the smart services. A model of the development process may also be useful in guiding how the dashboard should behave. The development model can be accessed through a service. For example, this notion of a smart dashboard object, which leverages both the user's model as well as the development process model, can be used to automatically show *mustfix* work items right before a release.

We anticipate that as dashboards become more prevalent that the user will need more tool support in specifying how it should be configured and how it should be updated over time. Currently such tool support is very limited. We suggest that some of the ideas that are currently being explored to support mashup creation by end users [24] could be applied to customizing dashboards. For more information on Mashups see the preceding Chapter by Grammel and Storey.

5.3 "Smarter" Feeds

For awareness on the basis of events, the Jazz CDE provides feeds. Feeds can either be displayed in the client application or as a viewlet as part of a web-based dashboard (discussed above). The most common way to view feeds is through the Team Central view in the client application. Figure 5 shows an example. Team Central is organized into multiple sections that are updated continually with the latest events. By default, Team Central displays a bar chart of current work for the signed-in developer by priority (see top of Fig. 5). The event log in the middle of the view's default configuration shows feeds. These are configured to include build events for all teams that the signed-in developer is part of, work item changes that are pertinent to the signed-in developer, and changes to teams.

Developers can add or remove feeds and filter events to personalize their event log, however, it can be cumbersome for the user to know which feeds and events to customize for their needs. In addition, incoming events are displayed as small popup windows in the client application, which may or may not be disruptive to the user.

A feed as described above is an example of a media channel rather than a media object. However, we suggest that a feed media object can be an instantiation of a user's query that can be delivered through various feed channels.

The *data, channels* and *representation* aspects of a smart feed object are very similar to those for the smart dashboard object (hence we do not repeat these descriptions here). In terms of *intelligence*, we believe the following behaviours could be supported by considering the user's model:
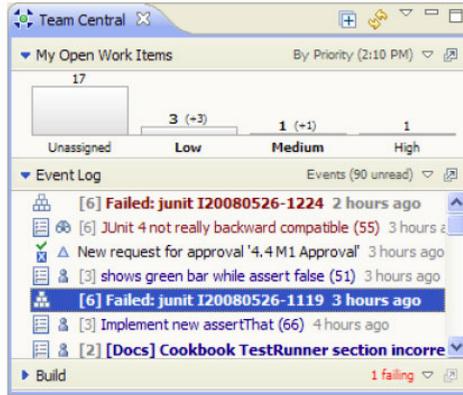
Fig. 5. Feeds in RTC Team Central.

- *Aggregation***:** Grouping of related events
- *Filtering* of events that are not of interest to the user (based on their task, phase in the process etc)
- *Relevancy***:** Ordering of events according to the user and process models
- *Immediacy***:** Choosing a channel based on urgency and user context
- *Consistency***:** Across channels

Adding "smartness" to feeds could help improve the delivery of important information to developers and also help to reduce the effects of information overload.   If the user is on email, then the smart feed can use that channel first to deliver important information, and once a user has read about an event through a feed, it should appear as "read" in other channels.

5.4   Summary: Towards a "Smarter" IDE

In this section we proposed experimenting with some of the ideas for smart media objects in the domain of software engineering. We gave three examples of how media objects could be constructed and described scenarios of how "smartness" that leverages smart interactions and smart services could improve developers' tasks. We suggest the domain of software engineering because developers tend to be early adopters of new technologies and because we have access to these users and to the tools that they use. Nevertheless, there are many challenges to be overcome. We discuss these challenges in the next section.

6   Discussion of challenges

The vision of a Smart Internet is an attractive one, but many challenges will have to be overcome to meet it. In this section we explore these challenges and pose them as future research questions.

6.1   Challenge #1: User's Model

It goes without saying that the smart internet, and therefore smart media and smart interactions will require a model of the user that documents the user's matters of concern, aggregates content that is important to the user (or at least aggregates access to it) and is situated within a systems view of the user's needs. Figure 2 in this chapter shows how the User's Model can be situated in the cloud and can play a role in the proposed architecture for the Smart Internet. This is one proposal, but the important question to ask is can a common user model be developed so that all services can use it? Will the community of service providers participate in developing the common model and adopt it? How resilient to change will the model be, and will the model support change where necessary? Matching the user's model to the user's mental model will be critical in providing appropriate cognitive support to the user, otherwise they may be distracted by disruptive *interruptions* or *misleading* recommendations.

A key question to consider, in terms of smart media, is what if all parts of the model are not known for a given user, will the services still work with incomplete models? When inconsistencies arise in populating the model, can these be handled?

6.2   Challenge #2: Session

Ng *et al.*'s vision of the Smart Internet [1], advocates for a concept of the user's session that is not bound to the server, but instead may crosscut multiple services and may also be asynchronous. But how will the smart internet determine what a user session is? Will the notion of a session be stored with the User's Model or with the Smart Media Object? From the Smart Media perspective, what information should a session include, what should it exclude? Where is this information stored, and will this new concept of a session be understood by legacy services?

6.3   Challenge #3: Collaborative Interactions

Although not discussed much so far in this chapter, we project that smart media objects will have to play a larger role in collaborative interactions. Thus these questions arise: How to model these collaborative activities across multiple users' models? How to detect social networks, that once known, could improve the services delivered to the individual and group users? Once collaboration comes into play, there are then many existing concerns about internet use such as security, privacy, awareness, etc, that have to be considered. For socio-technical concerns, we need to turn to the research area of Computer Supported Cooperative Work (CSCW) [25] for assistance.

6.4   Challenge #4: Rethinking Media as Smart Objects

In order to achieve the benefits we have proposed in this paper, it requires rethinking media artifacts as first class objects that have state, heuristics, and respond to and leverage smart services. This consequently means that changes may also need to be made to media channels and media objects. For example, if a change to a ticket is made when a user retrieves an old email message containing the ticket, the message needs to be updated somehow. Another issue not discussed so far is the level of

granularity of media objects. For example, should a media object contain all aspects of a travel booking such as flight, hotel and car rental? The level of granularity will in part depend on the user's model but the services that create the media objects will also need to determine this to meet the users' needs. Of course designing media objects that work well for users will require a lot of empirical work to understand how media artifacts are currently used, reused, shared and transformed.

Media objects that are "too smart" for the user's good is also at risk of damaging the user's experience – being smart ultimately means automating some steps on behalf of the user, but if the wrong steps are automated or if the right steps are automated in a bad way, this could be very disruptive for the user.

6.5 Challenge 5: Evaluating Smart Media and Smart Interactions

Another significant challenge may be how to study users when the way they use the internet services is rapidly changing. What may work one day, may not the next. More importantly, will we recognize characteristics of "smartness" when we see them? Borrowing ideas from McLuhan [26] on evaluation of media, we should consider: how will smart media *enhance*, make *obsolete*, *retrieve* and *flip* into when compared to social and other forms of media? Do smart media share many of the concepts of Boundary Objects [27], a concept from CSCW, which refers to how the same information artifacts are used in diverse ways by different communities? Will smart media objects be a form of boundary object? And if so, can we use evaluation techniques from the CSCW community to evaluate how smart media supports both individual and communities of users?

7   Conclusion

This chapter describes very preliminary thoughts on the concept of Smart Media, which we feel is needed to fulfill the vision of the Smart Internet of Smart Interactions and Smart Services. Our chapter does not offer technological solutions to the inherent challenges, but rather points to the socio-technical advantages that could arise if Smart Media did exist. In the meantime, there are many emergent activities in the research community that lean in this direction. Our goal is to study these paradigm changes, to take advantage of them, to learn from the past and hopefully help identify new opportunities for improving the user experience on the web.

References

1. Ng, J.W., Chignell, M., Cordy, J.R.: The Smart Internet: Transforming the Web for the User. In: Martin, P., Kark, A.W., Stewart, D. (eds.) Proceedings of the 2009 Conference of the Centres for Advanced Studies on Collaborative Research, CASCON 2009, Ontario, Canada, November 02-05, pp. 285–296. ACM, New York (2009)
2. Bush, V.: As we may think. The Atlantic Monthly (1945)
3. Nelson, T.H.: Complex information processing: a file structure for the complex, the changing and the indeterminate. In: ACM 1965: Proceedings of the 1965 20th National Conference, pp. 84–100. ACM, New York (1965)

4. Nielsen, J.: The art of navigating through hypertext. Commun. ACM 33(3), 296–310 (1990)
5. Tsihrintzis, G., Jain, L.: Multimedia services in intelligent environments: an introduction. In: Multimedia Services in Intelligent Environments, pp. 1–8 (2008)
6. Berners-Lee, T.: CERN (1989),
   http://www.w3.org/History/1989/proposal.html
7. Murugesan, S.: Understanding Web 2.0. IT Professional 9(4), 34–41 (2007)
8. O'Reilly, T.: What is Web 2.0: Design patterns and business models for the next generation of software (2005)
9. Raman, T.V.: Toward 2w, beyond web 2.0. Comm. ACM 52(2), 52–59 (2009)
10. Maybury, M.: Intelligent user interfaces: an introduction. In: IUI 1999: Proceedings of the 4th International Conference on Intelligent User Interfaces, pp. 3–4. ACM, New York (1999)
11. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM 40(3), 56–58 (1997)
12. Jameson, A.: Adaptive interfaces and agents. In: The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, pp. 305–330. L. Erlbaum Associates, Mahwah (2003)
13. Berners-Lee, T.: The Semantic Web Road Map. W3C Website (1998),
    http://www.w3.org/DesignIssues/Semantic.html
14. Brusilovsky, P., Kobsa, A., Vassileva, J. (eds.): Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, Dordrecht (1998)
15. Berners-Lee, T.: The year open data went worldwide. TED Talk
16. Park, S., Maurer, F.: The role of blogging in generating a software product vision. In: ICSE Workshop on Cooperative and Human Aspects on Software Engineering, pp. 74–77 (2009)
17. Reinhardt, W.: Communication is the key – Support Durable Knowledge Sharing in Software Engineering by Microblogging. In: SENSE 2009 (2009)
18. Begel, A., De Line, R.: Codebook: Social networking over code. In: Proceedings of ICSE, NIER Track (2009)
19. Kersten, M., Murphy, G.C.: Using task context to improve programmer productivity. In: SIGSOFT FSE 2006, pp. 1–11 (2006)
20. Van Deursen, A., Mesbah, A., Cornelissen, B., Zaidman, A., Pinzger, M., Guzzi, A.: Adinda: A knowledgeable, Browser-Based IDE. In: Companion Proceedings of the 32nd International Conference on Software Engineering (ICSE NIER). ACM, New York (2010)
21. McGrenere, J., Li, J., Lo, J., Litani, E.: Designing Effective Notifications for Collaborative Development Environments. In: Chignell, M., Cordy, J., Ng, J., Yesha, Y. (eds.) The Smart Internet. LNCS, vol. 6400. Springer, Berlin
22. Treude, C., Storey, M.-A.: How tagging helps bridge the gap between social and technical aspects in software development. In: Proceedings of the 31st International Conference on Software Engineering, pp. 12–22. IEEE Computer Society, Los Alamitos (2009)
23. Treude, C., Storey, M.-A.: Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, vol. 1, pp. 365–374. ACM, New York (2010)
24. Grammel, L., Treude, C., Storey, M.-A.: Mashup environments in software engineering. In: Proceedings of the 1st Workshop on Web 2.0 For Software Engineering, pp. 24–25. ACM, New York (2010)
25. Greif, I.: Computer-supported cooperative work: A book of readings. Morgan Kaufmann, San Francisco (1988)
26. McLuhan, M., McLuhan, E.: Laws of media: The new science. University of Toronto Press (1992)
27. Bowker, G.C., Star, S.L.: Sorting Things Out: Classification and its Consequences. MIT Press, Cambridge (2000)