

# Building for Social Translucence: A Domain Analysis and Prototype System

**David W. McDonald**  
The Information School  
University of Washington  
dwmc@uw.edu

**Stephanie Gokhman**  
Department of Communication  
Cornell University  
sbg94@cornell.edu

**Mark Zachry**  
Human Centered Design &  
Engineering  
University of Washington  
zachry@uw.edu

## ABSTRACT

The relationships and work that facilitate content creation in large online contributor system are not always visible. Social translucence is a stance toward the design of systems that allows users to better understand collaborative system participation through awareness of contributions and interactions. Like many socio-technical constructs, social translucence is not something that can be simply added after a system is built; it should be at the core of system design. In this paper, we conduct a domain analysis to understand the space of architectural support required to facilitate social translucence in systems. We describe an instantiation of those requirements as a system architecture that relies on data from Wikipedia and illustrate how translucence can be propagated to some basic visualizations which we have created for Wikipedia users. We close with some reflections on the state of social translucence research and some openings for this important design perspective.

## Author Keywords

Social translucence, system architecture, wikipedia.

## ACM Classification Keywords

H.5.3. Group and Organization Interfaces – Computer-supported cooperative work.

## GENERAL TERMS

Human Factors; Design.

## INTRODUCTION

Understanding and interpreting the behaviors of others in an online environment is hard. The cues and signals that we readily interpret in a face-to-face situation are not present or are at best attenuated. Lacking sufficient cues, users often misinterpret or misunderstand the actions and intentions of others. As the number of participants and the amount of interaction grows it becomes harder and harder for users to

make sense of others' actions, much less their own place in the community and the health of the community at large.

Social translucence [13,14] is a socio-technical term to describe how systems can facilitate understanding with regard to the actions of people in online environments. Social translucence includes three key design attributes: (a) mutual awareness of activities, (b) contextual propagation of socially salient cues (visibility), and (c) accountability for one's actions. Through support for these three characteristics members in a community can better understand the types of activities that transpire, understand the norms of the community and the consequences for the actions that they may take.

Like many socio-technical constructs, social translucence is not something that can simply be pasted in or bolted on. Designing for social translucence begins at the level of software architecture. The underlying software architecture of a system facilitates the observation of different types of activities within the system, the communications among the processes of the system, and the data that is stored and retrieved — all at the behest of some user action.

This paper raises the broad question of the domain of social translucence and conceptualizes it across two dimensions; actions that a system facilitates and levels of interpretation enabled by those actions. We start with a review of some relevant literature in social translucence. We conduct a type of domain analysis to elaborate these two dimensions of social translucence. Through our analysis we show that there is a wider space of social translucence yet to explore. We describe a system architecture that relies on data from Wikipedia enabling a broader class of social translucence, and we illustrate its applicability with specific use cases implemented using our architecture. The paper closes with a discussion and some implications for our contributions.

## SOCIAL TRANSLUCENCE IN SYSTEMS

Research to define and elaborate social translucence for systems derived from Babble [5, 12], a type of instant messaging system, and a subsequent version called Loops [16]. In these systems translucence was illustrated through a set of social proxies. A social proxy is a small visualization that represents activity or participation in some interactive online context [1]. For Loops and Babble,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CSCW'12, February 11–15, 2012, Seattle, Washington, USA.

Copyright 2012 ACM 978-1-4503-1086-4/12/02...\$10.00.

social proxies represented various forms of group communication, such as participants in a chat room or an online lecture. Social proxies can represent other forms of activity such as participation in an online auction or individuals queuing up for a chat with an online technical support specialist [14]. In this way, social proxies are a form of social visualization.

The term social translucence and techniques that can be broadly construed as supporting social translucence have been applied to wikis, often in the case of Wikipedia. History Flow [10, 33, 34] visualizes the changes of wiki pages over time, which is directly related to the activities of the editors. One claim is that, through a History Flow visualization, a user can see an ‘edit-war’. We point out that one actually sees a series of changes being removed and replaced. That this series of actions is an ‘edit-war’ is a social interpretation that is made by an observer who may know something special about the community for which History Flow is visualizing activity.

Wikipedia has seen the development of numerous social visualizations such as Chromograms [35], Revert Graph [22], WikiDashboard [30], WikiChanges [27], and Revision History [11]. Some of these were deployed in live or proxied versions of Wikipedia meant for user consumption [30], while others were stand alone or analytic visualizations designed mostly for research.

The majority of these prior visualizations are based on ‘surface’ data. Here we are defining ‘surface’ data as data that can be visualized without little addition transformation. That is, the action of an edit and who made an edit may be all that is necessary to create the visualization. There is no need to transform these surface actions into some other representation before a meaningful visualization can be created. The actions are likely to be meaningful in their own right. The idea that this is a good way to develop social translucence is supported by a short list of claims that Erickson proposed in later work [15].

A key exception to this is the Chromograms visualization [35]. This visualization processed revision comments, giving each comment a consistent hue, saturation and brightness based on the first three letters of the comment string. While this is not very deep, it moves one step beyond a surface representation that an edit happened or that an edit has a comment. Interestingly, this simple processing of comments demonstrated activity patterns like the systematic processing of article pages needing cleanup or categorization.

While much of the related work in social translucence as applied to wikis implicitly equates the need for visibility with a type of (often complex) social visualization, this is not a requirement of social translucence. Another of Erickson’s claims is that the visualizations should be simple and open to interpretation [15]. This claim has had some

support in at least two cases [13, 31]. This claim is the root of the distinction we drew above for History Flow between what is visualized and the potential interpretations of that visualization.

Social translucence has also appeared in the threads of research concerned with awareness and availability to interruption. In this thread, visualizations are one mechanism for providing awareness of patterns of work activity in order to gauge levels of availability [3]. Some systems like Lilsys [4], MyVine [18], and AvBox [31] went further to include a form of environment sensing, and often manifest this awareness through visualizations. These sensing applications often required more processing. That is, they relied on some type of inference engine operating on samples of the physical or virtual environment to determine what should be conveyed through the visualization.

Much of the systems work in social translucence focuses on the level of the interface through visualizations. Many of those visualizations are fundamentally stand-alone features that overlay an existing system. In many cases these visualizations are re-representing activity in the system that occurs at the surface. A few systems have relied on sensing and inference algorithms to infer that activity (or lack of activity) should be visualized.

We extend the existing work in social translucence by framing social translucence on two dimensions through a type of domain analysis. The analysis opens up a space of possible implementations. We begin exploring the space of implementations through an architectural extension to WikiMedia and illustrate some of that space through specific use cases. We first turn to our domain analysis.

### **SOCIAL TRANSLUCENCE: A DOMAIN ANALYSIS**

A proper domain analysis would consider a full range of software that claims to support social translucence to understand the specific features and architectures that have enabled those systems. In the space of systems that claim to address social translucence, there is little work focused on the software architecture. Instead, most of the systems described exhibit social translucence through somewhat independent software features, many of which we have covered above.

As well, a strict domain analysis focuses on software and architectural aspects from a technical stance. Since social translucence is a socio-technical construct, taking either a purely technical or purely social stance would ignore the important interplay between social activity and the systems that facilitate that activity. In our domain analysis we consider both the prior technical systems and motivating examples of activity in contributor systems.

We draw our motivating examples from the range of activity that happens in wikis and specifically Wikipedia.

The keystone papers in social translucence often motivated their explanations with examples based on a chat or instant messaging (IM) system. Wikis share some of the properties of IM, but afford a potentially wider range of user activities — depending on the wiki.

### **The Domain of Social Translucence**

While there has been work developing socially translucent visualizations, there has not been work to understand the broader space of social translucence. Our efforts to design an architecture that facilitates social translucence forced us to face this problem and attempt to analyze the space of social translucence. Taking a step back from the instantiations of social translucence that we see in the literature, there are two key aspects from the definition of social translucence that could influence an architecture that would support it: user action and contextual salience.

Much of the ‘social’ in social translucence is based on actions that are oriented toward others in an interactive context. In many collaborative systems those interactions are mediated by the system and can often be decomposed into discrete actions with the system that are later shared with some partner, either synchronously or asynchronously. At times a system might not even preserve strict temporal ordering among the sets of discrete actions taken by a distributed set of people.

Breaking down the notion of ‘social’ in this way is important to thinking through the range of actions that might form the basis for a socially translucent system. In short, from an architectural perspective, almost any action that a person might take with a system could be an opportunity for a mediated interaction. Thus, almost any action with the system is the basis for generating social translucence. Whether or not a system exposes that an action has happened is important for social translucence at an architectural level.

Contextual salience is another critical element of social translucence. The challenge architecturally is that a system implementation cannot possibly anticipate all of the salient contexts in which actions might be represented. Yet, in all systems, for an action to be possible in the system there must be some code that understands and implements the action whether it be a button press, a drag, a GET or POST request, or any myriad other types of actions. These actions are not simply for collecting, they form the frame for the users’ continuing activity. A socially translucent system needs to account for this ongoing activity when presenting information to the user or taking actions.

Our domain analysis is framed by two dimensions; one related to types of user actions in collaborative systems, and a second describing the processing and interpretation done by the system. Loosely, the first dimension begins to layout the types of actions that we might want the system to know and reflect about individuals. The second dimension relates

to the types of interpretations that the system could make. These system interpretations are not the same as those made by users. Instead they represent aspects of behavior that might be contextually propagated for users.

### *Action Dimension*

The first dimension contains four categories of actions that we term Content, Interaction, Relation and Systemic. These categories are not arbitrary: they follow from the types of questions individuals want to answer about others’ behaviors [8, 9, 10].

**Content** - Understanding and representing the contents of a user contribution is a key aspect of representing social translucence. Content covers a wide range of things such as text, pictures, sound, video as well as meta-data. The contribution of content is valuable in some contexts, but in many cases the architecture should represent an additional level of information, indicating what that content was about. Wikis tend to be content focused and in many wikis a change history makes every piece of content visible and inspectable. In Wikipedia, knowledge of the content can be used to infer attributes of possible future contributions and how they might be trusted [2].

**Interaction** - Collaborative systems are often about the types of interactions that individuals can have through the system. Exchange of messages are an obvious form of interaction, but user interaction can be mediated by systems in a number of more complex ways. The domain of social translucence should account for each user action that contributes to the interactions among users. Wikis have some properties that are similar to IM in that there are explicit places for conversation, even if they are not synchronous. In Wikipedia individual user pages and user talk pages are explicit locations for attempting interaction. But co-authoring and reverting edits have also been used to illustrate a type of mediated interaction between and among users [22, 32, 34]. Not all co-authoring or reverting actions are reciprocated. Indeed notions of what constitutes an interaction can vary as a function of the actions that individuals take and the social context in which those actions are taken.

**Relation** - Not all relations are a function of interactions. In collaborative systems users can articulate specific relations that might need to be represented as a form of social translucence. For example, the explicit ‘friending’ of one user, which may not be reciprocated, is an important action. Also, the act of tagging an item creates a relation between that item and other items that also carry that specific tag. Collecting, organizing and presenting relations among people and things are valuable for social translucence. Link or relation analysis has shown value for understanding web influence such as the use of PageRank and broadly in understanding social structure such as in the use of Social Network Analysis. Too often, a system reflects little of who

<b>Structure</b>	Structure is the higher level constructs like roles and norms that individuals can observe over time. Structures are made of sequences of Instances and Series. Since Social Translucence is a contextual entity, time periods necessary to observe structure are defined by the social context.			
<b>Series</b>	Series are comprised of Instances. However, a Series may include more than one type of Instance. Common series can be comprised of repetitive activities, such as a chat where individuals trade interaction Instances.			
<b>Instance</b>	Actions that contribute content, but not simply the fact of a contribution. A system can process the content to see what type it is, some meta data on the content, or what the content is.	A single action by one user oriented toward another user. An attempt to talk or post a reply is an attempt to interact by one user, but so is a "poke" or a barnstar award. An Instance of interaction does not have to be reciprocated.	Users create explicit relations through different actions. For example, "friending" is a single action that attempts to indicate a relation. As well, tagging an item creates a relationship among the newly tagged item and all other items that are tagged the same.	Actions that change the status of users in the system or the state of the system. For example, login, banning, user permission granting, programming, or running a bot.
	<b>Content</b>	<b>Interaction</b>	<b>Relation</b>	<b>Systemic</b>

**Table 1. Two dimensions of the social translucence domain analysis**

created the relations nor anything about the context in which those relations were created.

**Systemic** - Systemic actions are those that change the state of the system and the status of users of the system. For example, many systems require a 'login' action, but not all of them represent that action publicly and make it available broadly. 'Login' is social in the sense that it moves a person from a social place of unknown outsider to a known or at least pseudoanonymous community member. In virtual worlds or 'space' based systems, the location of the user is often a systemic trait of a user. There are other systemic actions that are key to facilitating social translucence in a system. In Wikipedia the Request for Adminship (RfA) process engages the community in a review of candidate users to potentially grant additional permissions. This process has implications for the community, but the actual granting of permissions moves the user from one status to another. Status changes in the system are important for understanding what is happening socially in a complex collaborative system.

#### *Levels of System Interpretation*

The second dimension of our domain analysis consists of three levels of interpretation that a system might distill to represent actions that users take within the system. We define these three levels as *Instance*, *Series*, and *Structure*. Each level builds on one or more levels below it. The level of Instance is the basic level of representing that a user has taken some individual action. In the common example of a chat system, a *Systemic* Instance is that the person joined the chat, a *Content* Instance is the text of one contribution, an Instance of *Interaction* is **that** they contributed to the conversation without presenting the actual content, and a *Relation* Instance could be establishing a private conversation with one other member of the chat. These instance examples all have real world analogs for what we may see in public as a function of the translucency of a social setting (e.g., [20]).

Using a wiki-based example, a *Systemic* Instance is a status change, like being promoted to admin. A *Content* Instance

is editing an article. An *Interaction* instance would be posting to a user talk page or an article talk page to try and engage in a conversational interaction. Lastly, in wikis, tagging and categorizing pages constitutes a *Relation* Instance activity.

At the second level, we define *Series* as repeated Instances, whether those instances are of a single type or a heterogeneous patterned set. Series are important because they reflect how individuals participate and act in consistent or patterned ways and how they take on key tasks within a complex collaboration. Series suggest behaviors and interactions that might be sustained. The meaning of a Series is open to interpretation. For example, one user might perform a Series by systematically vandalizing a set of wiki pages, or an admin might perform a series by moving through a set of administrative tasks such as sanctioning users. Series are an aggregation of Instances. For a system to identify Series it must maintain some large set of Instances for some period of time. Further, once a Series is recognized by the system, the knowledge that the Series has occurred needs to be made available and potentially stored. If Instances are stored for long periods of time, like in many wiki systems, then storing a Series may simply require the system to recognize that a Series has happened based on collected Instances.

*Structure* is our third level of meaning and interpretation. There are many ways that communities structure their social space. Expectations, normative behaviors, patterns of social interaction, and social roles are some of the types of structure that individuals in a community might observe.

We define a Structure as some combination of recognized Series and Instances over time. While Series do occur over time in the sense that the first Instance in a Series naturally comes before the last Instance in a Series, a Series would not necessarily let one recognize a Structure. Our notion of Structure specifically requires time before a Structure could be acknowledged, but we leave the specific notion of time open to interpretation and as a function of the social context in which social translucence would be conveyed. Similar to Series, we consider Structure to be composable over

existing Series and Instances. We frame the proposition for our domain analysis by claiming that some combination of Series and Instance would be necessary for a Structure to be observed. Further, we open the question of how to make a Structure visible for interpretation by individuals in the community.

Our domain analysis opens up the possible implementation space of social translucence. Next we describe a software architecture that enables wider exploration of social translucence and illustrate some of that space through several social proxies.

### RE:ARCH – AN ARCHITECTURE SUPPORTING SOCIAL TRANSLUCENCE FOR WIKIS

Our domain analysis raises architectural concerns that must be instantiated in some fashion in order to support social translucence. We have built an extension to Wikipedia that we call Re:Arch which allows us to explore how social translucence can be enabled at the architectural level. We implemented Re:Arch through proxying because we ultimately want to facilitate actual user interactions and we have no control over when or how new code is incorporated into the WikiMedia codebase. This approach is similar to other single use tools [23, 30].

The general architectural model we employ parallels the data centered architecture of many contributor systems. We describe Re:Arch as a type of loose blackboard architecture (see Figure 1). In software architecture, blackboards are a generic architectural style sometimes referred to as a “repository” or “data-centered” style. We call this a blackboard because this architectural style facilitates complex state sharing, while loosening the constraints of the formal blackboard system [29]. Initially “blackboard” defined a strict rule-based control structure [21], but modern adaptations allow for alternate control flows to influence the state of the data on the blackboard. We follow this architectural style as a complement to many types of contributor systems. Versions of our architecture could simplify the construction of a range of tools and visualizations that support social translucence.

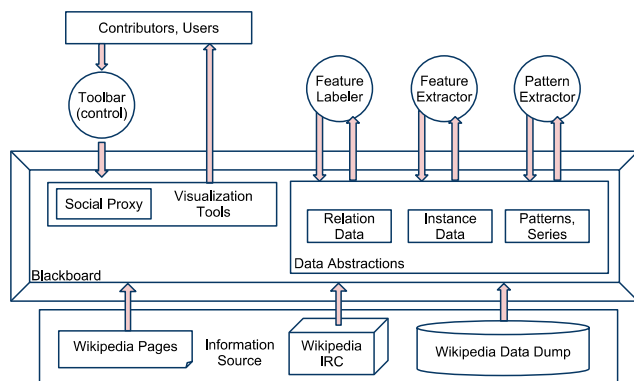


Figure 1. Re:Arch architecture diagram.

In addition to the core blackboard, Re:Arch includes visualization tools that allow user actions to alter the state of a blackboard: extracting data, designating patterns within the data, and posting this data in a new form.

### Blackboard

The blackboard weaves together content, interaction, relations and system state. The blackboard has processes that inspect data from an information source, performs abstractions on this processed data and then utilizes this information to update the visualizations seen by the users. This information manifests itself within several different components that comprise the blackboard.

The blackboard provides a composable space in which each of these components can be manipulated by a contributor or user. Logically the blackboard enables users to see the entire state of the system at multiple layers; user, content, set of content, etc. A contributor can elicit meaningful information in their choice of context from separate but interrelated tools that are offered within the blackboard. These contexts are determined within the toolbar (or control) features of the architecture.

The data layer of Re:Arch instantiates key aspects from the domain analysis. A *feature extractor* reads Wikipedia pages, processes and extracts features from Wikipedia page dumps. The feature extractor can simply collect surface level features such as edits by given users, but it also processes the data to collect features that are not at the surface. In this way our Instance level subsumes what we previously defined as surface data. For example the feature extractor can collect headings, section contributions, template tags, links, and barnstars. The feature extractor implements much of what is needed at an *Instance* level.

Building from the feature extractor, the *feature labeler* annotates raw features in order to begin developing meaning from raw features. These labels are generated through a higher-level processing such as machine learning, crowdsourcing or human computation. The labeling is conducted in a contextually dependent way. One example is the awarding of barnstars on Wikipedia. Barnstars are small acknowledgments or achievements added to a user’s talk page by another user. The feature extractor may identify a number of barnstars for a given user, and the feature labeler could label those barnstars with the type of work being acknowledged. It is up to user interpretation to determine whether the barnstars were awarded for negative or positive contributions. The feature labeler begins to address how interpretations can be presented that move from *Instance* to *Series* to *Structure* levels.

Building upon the feature extractor and labeler, the *pattern extractor* looks for patterns or chains of activity. Patterns could be mined for an individual user or for groups of users, such as WikiProject participants. The pattern extractor identifies and collects possible patterns from existing

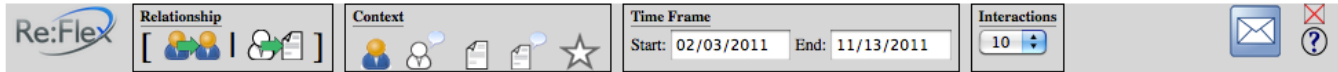


Figure 2. Re:Flex toolbar controls the composition of data sources and time window for the social proxy visualizations.

features or feature labels on the blackboard and posts them back. The pattern extractor is a direct instantiation of how Re:Arch identifies candidate *Series*. Identifying *Series* is valuable for users of the system and for understanding patterns of participation in open collaboration.

Re:Arch includes social proxies which rely on the visualization tools and data from the data layer. Our social proxies, much like those in the original conceptualization of social translucence, are not simply one visualization, but a range or set of possible visualizations. Social proxies can compose results from any information in the data layer of the blackboard.

### Information Source

The system draws raw data from a number of sources to feed the blackboard. While our specific example draws on Wikipedia, the information source represents the logical connection between our architecture and potentially other user generated content systems that would want to implement social translucence.

The information source for Re:Arch is an aggregate itself of a number of Wikipedia data sources including Wikipedia Data Dumps, Wikipedia Special Pages and Wikipedia IRC Streams. The Wikipedia Data Dumps are the regular dumps of the entire edit history of Wikipedia from inception. This data set is cleaned and processed in order to populate our system's data abstraction layer. We can use the Special: page mechanism in Wikipedia to request specific updates, or make specific request from various Wikipedia tools.<sup>1</sup> Wikipedia also supports several Internet Relay Chat (IRC) channels where page changes are announced within the chat text. The architecture includes a process that listens to one or more channels and posts updates to the blackboard.

### Toolbar as Control

A toolbar from the visualization toolkit facilitates a set of actions within this architecture. The toolbar provides the options to select data sources and present those sources in a number of different ways. The toolbar exists as a form that delivers specifications to the blackboard to pull meaningful information for the user. These specifications include: time window, types of data requested, and number of relations<sup>2</sup> to include in the visualization. In the following section we describe more about how the toolbar works, and give some examples of visualizations.

1. <http://en.wikipedia.org/wiki/Wikipedia:Toolserver>

2. This relation should not be confused with our analytic use of relation. The use here in the paper and in the toolbar is the general notion; a relationship between items.

### RE:FLEX – INTERACTIONS AND VISUALIZATIONS

Re:Flex is a set of interface components built as part of Re:Arch. Re:Flex supports social translucence in Wikipedia by enabling editors to arrive at relatively complex understandings of other editors by visualizing interactions among editors or between editors and content. The system visualizes these Series that prior research has illustrated are meaningful to the Wikipedians (e.g., [6, 7, 9, 19, 22]).

For example a qualitative study by Bryant et al. [6] illustrated that newcomers to Wikipedia become part of the community through observing and modeling the behaviors of others in the community. This is not to dismiss the value of instruction, coaching, and sanctioning for shaping participants' behaviors, just that observation is a key aspect. This suggests that observing a Series maybe important to enabling newcomer participation.

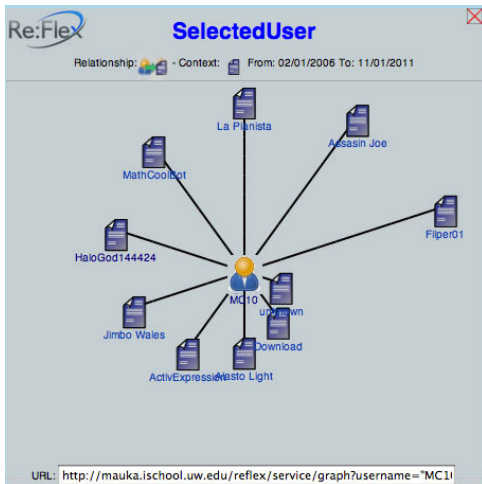
Some quantitative work by Burke & Kraut [7] focused on the Request for Adminship (RfA) review process, wherein regular Wikipedia editors who successfully complete the process are granted additional administrative capabilities. Their analysis resulted in a quantitative model of the user characteristics that account most for a positive outcome of the RfA process. While the model provides a score on how RfA candidates are judged, users are likely to want more than a composite score for a candidate — they may want to investigate that candidate in other ways.

In some initial work we did on the RfA process [9] reviewers wanted to know whether a candidate 'hung out with' the right kind of people on Wikipedia, and whether the candidate was participating in the right places. The subtlety is that for different reviewers what makes 'right' is open. Different reviewers value different 'right' people and different 'right' locations. This is reinforced by some exploratory design work of Kittur et al. [22] that investigated user revert graphs to visualize which users were in conflict with each other as a function of reverting each others' changes.

Re:Flex was developed to enable further exploration of social translucence as framed by our analysis (see Table 1). The current implementation is focused around supporting interpretations of Series. Little emphasis in this version was placed on visualizing Instance, and future work will focus on supporting the Structure level of interpretation.

The Re:Flex toolbar (Figure 2) floats over Wikipedia pages and offers users the opportunity to visually explore different kinds of relationships. While working in Wikipedia, users can turn on or off the toolbar, depending on the kind of work they are engaged in. When they want to know





**Figure 3. Example Re:Flex editor-article relation graph. Target (hovered) username presented in the middle, with top 10 edited articles placed by relative frequency on the edges.**

something more about other editors, they can turn on Re:Flex and visually explore activities of another editor.

Use of the toolbar entails selecting the desired attributes of the target editor that will be included in the visualization. The set of visualizations that we describe allow the exploration of relationships between editors and relationships between editors and artifacts. These relationships are an example of taking Instance data (an edit) and transforming it to Series (a relation). The user may choose to expose editor-centric relations based on data from one or more of the four data contexts: article, article talk, user or user talk.

With one or more of these contexts selected, Re:Flex includes the relevant data in a visualization that is generated when the user hovers over an editor's username. The user has further control of what is exposed in the visualization by defining the date range of attributes to be included in the visualization and the number of top relationship entities to be visualized. For example, the user can select a target date range, say 3 months, for a given year, and specify the desired number of top interactions in any or all of the categories. Re:Flex will generate a visualization in a pop-up window located near a hovered editor's username. This visualization window persists until the user closes it or until the user navigates to another page. The user may generate several of these visualizations on a page, selecting the desired categories and hovering over different editor names to compare the visualizations.

Beyond these interaction categories, the Re:Flex toolbar also includes a feature for visualizing significant work and awards. This visualization is based on the barnstars or other significant recognition from the Wikipedia community. Unlike the other selections, this editor attribute cannot be combined with the other interaction categories. That is,

when the recognition (star) icon is selected, the other interaction category icons are all deselected, and the user sees a visualization based on a hovered editor's barnstar acknowledgments. These are generated by processing the text of the barnstars that awarded to the target editor over the time period specified by the user. Details about how this awarding text is parsed, interpreted, and articulated to different categories of work is described elsewhere [26, 28].

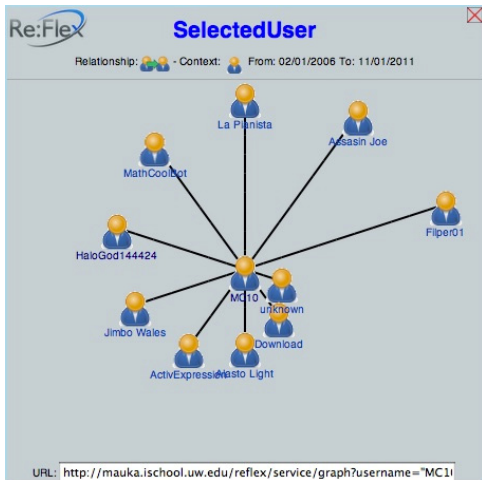
The five types of visualizations that can be generated in this version of Re:Flex are motivated by use scenarios that we have identified from prior research with Wikipedians. These visualizations and the use scenario described below represent a small set of the possible ways that Re:Flex can support social translucence in contributor systems.

### Re:Flex Example Scenarios

To illustrate how the Re:Flex system supports social translucence, we briefly point to one application scenario. We consider how Re:Flex enhances user understanding in the Request for Adminship (RfA) process in Wikipedia. The RfA process is a structured process wherein Wikipedia editors consider nominated members of their community for potential promotion to admin status [7]. Participants in this high-stakes process are encumbered with responsibility for making relatively sophisticated judgments about their peers, but provided with limited resources for doing so [9].

Consider Katie, an active Wikipedia editor. Katie stopped by her user page and saw a note from a friend saying that she should look at an ongoing RfA case. Katie has participated in an RfA review before. She believes in holistic evaluation of the candidate, to include edits to article pages, work with policy, removal of vandalism, and congenial interactions with other editors. In the past this has required that she plow through page after page of revision history. On the RfA page she sees a new tool, one that she has not seen before: Re:Flex. Since making article contributions is an important aspect to being a good Wikipedia editor, Katie selects article pages (document icon in Figure 2) as her initial exploration with the tool. She hovers over the candidate's username and a window pops-up showing the top five articles edited by that editor (Figure 3). These articles are good starting places for understanding how this individual contributes content.

But Katie also wants to understand with whom the candidate has interacted. She selects the user data source (user icon in Figure 2) and hovers over the username of the editor. A window pops-up (Figure 4) showing the relative relational strength between the candidate and the top five other editors with whom the candidate has interacted. Katie sees relationships rather than simply edit counts. These could be the key relationships that determine whether the RfA candidate interacts well with others and is worthy of adminship.



**Figure 4. Example Re:Flex editor-editor relation graph. Target (hovered) username presented in the middle, with top 10 related editors spaced by relational strength.**

Katie sees another option in Re:Flex, the work and award type of interaction (star icon in Figure 2). Katie clicks the star and hovers over the RfA candidate's username. A window opens containing a word cloud illustrating the relative weight of acknowledgements across seven broad categories of wikiwork; Administrative, Border Patrol, Collaborative Action, Editing, Meta-Content, Social and Community, and Miscellaneous (Figure 5). The word cloud conveys relative weight of types of work rather than a simple count of the barnstars or achievements. Based on the types of work, Katie can decide whether to explore one or more specific areas of Wikipedia edit history to better understand how this candidate contributes to Wikipedia.

RfA tools currently supported in Wikipedia are low level tools, mostly providing counts of content contributions often organized by namespace. Re:Flex addresses the relative paucity of information available to editors during this deliberative process by providing visualizations that go beyond just edit counts.

While RfA is illustrative of a formal Wikipedia process that might benefit from social translucence, there are many other, less structured places in Wikipedia where social translucence can be useful for the community, particularly, the desire among Wikipedians to know something more about the other editors they encounter. This desire is based, at least in part, on to the usefulness of understanding the intentions of others in order to react accordingly. Is this person serious? Is this person engaged in malicious behavior? Does this person need help or censure? Through Re:Flex, editors can develop some insight into other editors, discovering where another has contributed to the system.

### DISCUSSION & IMPLICATIONS

Large-scale contributor systems present a special domain for considering social translucence. Applying our analytical

framework for social translucence to these large-scale collaborative systems opens the space of socially translucent designs and implementations. Making visible the contextually salient work of users in these systems will enable a better understanding of each other's activities as well as facilitate productive work.

Based on our framing, it is not difficult to imagine redesigns of prominent sites to expose more of the users' contextual Interactions, Series, and Structure. We recognize that greater social translucence is not uniformly positive. For example, enabling more translucence in a social networking site would shift the nature of the system in a notable way. What if users' social searching and social browsing [25] were explicitly exposed? What if users' patterns of engaging in chat conversations about postings of others were made visible? Series and Structure insights like these could make social networking more socially translucent system, but might not complement existing uses and culture. Unlike a social networking site, contributor systems would generally benefit from greater social translucence. The point is that designers need to make informed decisions about where to use translucence. Our framework allows designers to make informed decisions about what activities to visualize at what interpretive levels.

The two-dimensional view of social translucence we have developed holds promise beyond providing heuristics for designing translucence into contributor systems. The definitional properties of the two dimensions — types of actions and the processing/interpretation done by the system — can also be operationalized as analytic categories for assessing the nature and prominence of social translucence in collaborative systems more generally. Categorizing the online systems that people use to work together on the web, for example, has proven to be difficult because such systems are varied in multiple ways [17].



**Figure 5. Example Re:Flex award and recognition word cloud visualization. The word cloud displays the relative amount of recognition of an editor for a set of defined work types.**



Given that all such systems share the fundamental characteristics of facilitating multi-participant interactions mediated by web-based tools, it would be possible to categorize this class of technology using our dimensions of social translucence. Such a structured classification of socio-collaborative systems would provide some reasoned order to the confusion of systems that are now labeled social networking sites.

Our contribution goes beyond just our intellectual framing of social translucence. Through Re:Arch and Re:Flex we provide a concrete instantiation of some properties derived from the framing. Re:Arch supports social translucence through the data on the underlying blackboard, through the processes that transform and label that data, and through the social proxy visualizations that we have designed.

Many of the prior system contributions that support social translucence are focused at the Instance layer, or individual actions with the underlying system. However, Instance based visualizations are only a small portion of the realm of social translucence. Building off the Re:Arch architecture with Re:Flex, we go beyond a single visual design toward validating the theoretical framework of social translucence at the Series level. Building Re:Flex on an architectural model designed to facilitate social translucence enables wider exploration of translucence at the interface.

The interaction category relationships present in the current Re:Flex prototype are relatively simple, based primarily on interaction counts. These were chosen because they align with existing Wikipedian practices and because few other tools address these needs. These visualizations suggest ways of exploring Series-based insights about others, but do not fully realize that interpretation layer of our analysis. We do not mean to imply that interpreting the existence of a relation based on Series is simple and solved.

In Re:Flex user-to-user relations can be requested of the blackboard in more than one way. A user-to-user relation is one way to visualize a Series of edits. But there are different ways to think about those individual edits. For example, the co-editing of an artifact is one way to recognize a potential user-to-user relation. But also, in the context of a user's page, which is a type of 'owned' artifact, a unidirectional edit might be sufficient to recognize a potential relation. Further, in that same context one might want to consider reciprocal edits, where users mutually edit each other's user pages. Yet further, the different mechanisms for considering the edit action might be combined — edit, co-edit, and/or reciprocal edit — to distinguish further ways of recognizing a potential user-to-user relation.

The award recognition visualization is built from more complex, Series-based observations about a given editor, thereby addressing a different aspect of our framework. The award recognition visualization relies on a multi-label

machine classifier. Recognitions are then compiled over time as they are awarded. This feature based visualization is just an example and validating the utility of the visualization is future work.

Effective social translucence relies on mutual awareness of activities, contextual propagation of socially salient cues, and accountability for one's actions. Our analysis helps designers think more concretely about what activities are important to social translucence and how interpretive levels of those activities might provide contextual salience.

## ACKNOWLEDGMENTS

We acknowledge the contributions of our collaborators, specifically Michael Gilbert, Casey Shey, Cheng Chuang, Alena Benson, and Siwei Kang for their work on Re:Flex. We thank the CSCW reviewers for an engaging discussion on our work and for the resulting improvements. We thank Wikipedia users for sharing their expertise and WikiMedia foundation for providing dumps of this valuable resource. This material is based on work supported by the National Science Foundation under Grant No. IIS-0811210. The opinions, findings, conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

1. Ackerman, M.S. and Starr, B., Social Activity Indicators: Interface Components for CSCW Systems. *Proceedings of UIST'95*, 1995
2. Adler, B.T. and de Alfaro, L. A Content-Driven Reputation System for the Wikipedia. in *Proceedings of WWW 2007*. 261-270.
3. Begole, J., Tang, J.C., Smith, R.B. and Yankelovich, N., Work Rhythms: Analyzing Visualizations of Awareness Histories of Distributed Groups. *Proceedings of CSCW 2002*.
4. Begole, J., Matsakis, N.E. and Tang, J.C. Lilsys: Sensing Unavailability. *Proceedings of CSCW 2004*.
5. Bradner, E., W.A. Kellogg, and T. Erickson. The adoption and use of BABBLE: A field study of chat in the workplace. *Proceedings of ECSCW 1999*.
6. Bryant, S.L., Forte, A. and Bruckman, A., Becoming Wikipedian: Transformation of Participation in a Collaborative Online Encyclopedia. *Proceedings of GROUP 2005*. 2005.
7. Burke, M. and Kraut, R. Mopping Up: Modeling Wikipedia Promotion Decisions. *Proceedings of CSCW 2008*. 27-36.
8. Carley, K. M. A Dynamic Network Approach to the Assessment of Terrorist Groups and the Impact of Alternative Courses of Action. In *Visualizing Network Information*, Meeting Proceedings RTO-MP-IST-063 available at <http://www.rto.nato.int/abstracts.aspx>

9. Derthick, K., Tsao, P., Kriplean, T., Borning, A., M. Zachry, M., and McDonald, D. W. Collaborative Sensemaking during Admin Permission Granting in Wikipedia. *Proceedings HCI International*. 2011
10. Diaz, O. and Puente, G. Model-aware Wiki Analysis Tools: the Case of HistoryFlow. *Proceedings of WikiSym 2010*.
11. Ekstrand, M.D. and Riedl, J. rv you're dumb: Identifying Discarded Work in Wiki Article History. *Proceedings of WikiSym 2009*.
12. Erickson, T., Smith, D.N., Kellogg, W.A., Laff, M., Richards, J.T. and Bradner, E. Socially Translucent Systems: Social Proxies, Persistent Conversation, and the Design of "Babble". In *Proceedings of CHI'99*.
13. Erickson, T. and Kellogg, W. A. Social Translucence: An Approach to Designing Systems that Support Social Processes. *ACM Transactions on Computer-Human Interaction*, 2000. 7(1).
14. Erickson, T., Halverson, C., Kellogg, W., Laff, M. and Wolf, T. Social Translucence: Designing Social Infrastructures that Make Collective Activity Visible. *Communications of the ACM*, 2002. 45(4).
15. Erickson, T., Designing Visualizations of Social Activity: Six Claims. *Proceedings of the CHI extended abstracts (CHI'03)*, 2003.
16. Erickson, T., Kellogg, W. A., Laff, M., Sussman, J., Wolf, T. V., Halverson, C. A., and Edwards, D. A Persistent Chat Space for Workgroups: The Design, Evaluation and Deployment of Loops. *Proceedings of DIS 2006*.
17. Ferro, T., and M. Zachry. (2011). Networked Knowledge Workers on the Web: An Examination of US Trends, 2008-2010. *Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions*. Hershey, PA, USA: IGI Publishing.
18. Fogarty, J., Lai, J., and Christensen, J. Presence versus Availability: The Design and Evaluation of a Context-Aware Communication Client. *International Journal of Human-Computer Studies (IJHCS)*, 2004, 61(3).
19. Forte, A. and Bruckman, A. Scaling Consensus: Increasing Decentralization in Wikipedia Governance. *Proceedings of HICSS-41*, 2008.
20. Goffman, E. *Behavior in Public Places*. Free Press, New York, NY, 1963.
21. Hayes-Roth, B. A blackboard architecture for control. *Artificial Intelligence*, 1985, 26, 251-321.
22. Kittur, A., Bongwon, S., Pendleton, B.A. and Chi, E. He Says, She Says: Conflict and Coordination in Wikipedia. *Proceedings of CHI 2007*.
23. Krieger, M., Stark, E. and Klemmer, S.R. Coordinating Tasks on the Commons: Designing for Personal Goals, Expertise and Serendipity. *Proceedings of CHI 2009*, 1485-1494.
24. Kriplean, T., Beschastnikh, I., and McDonald, D. W. Articulations of Wikiwork: Uncovering Valued Work in Wikipedia through Barnstars. *Proceedings of CSCW 2008*.
25. Lampe, C., Ellison, N. and Steinfield, C. A Face(book) in the Crowd: Social Searching vs. Social Browsing. *Proceedings of CSCW 2006*.
26. McDonald, D. W., Javanmardi, S. and Zachry, M. Finding Patterns in Behavioral Observations by Automatically Labeling Forms of Wikiwork in Barnstars. *Proceedings of WikiSym 2011*.
27. Nunes, S., C. Riberio, and G. David. WikiChanges - Exposing Wikipedia Revision Activity. *Proceedings of WikiSym 2008*.
28. Sajnani, H., Javanmardi, S., McDonald, D.W. and Lopes, C. Multi-Label Classification of Short Text: A Study on Wikipedia Barnstars. in "Analyzing Microtext" Workshop at the Twenty-Fifth Conference on Artificial Intelligence (AAAI-11), 2011.
29. Shaw, M. and Garlan, D. *Software Architecture: Perspectives on an Emerging Discipline*, 1996. Prentice-Hall, Inc.
30. Suh, B., Chi, E., Kittur, A. and Pendleton, B.A. Lifting the Veil: Improving Accountability and Social Transparency in Wikipedia with WikiDashboard. *Proceeding of CHI 2008*.
31. Szostek, A.M., Karapanos, E., Eggen, B. and Holenderski, M. Understanding the Implications of Social Translucence for Systems Supporting Communication at Work. *Proceedings of CSCW 2008*.
32. Tang, L.V.-S., Biuk-Aghai, R.P. and Fong, S. A Method for Measuring Co-authorship Relationships in MediaWiki. *Proceedings of WikiSym 2008*.
33. Viégas, F., Wattenberg, M. and Dave, K. Studying Cooperation and Conflict between Authors with history flow Visualizations. *Proceedings of CHI 2004*, 575-582.
34. Viégas, F., Wattenberg, M., Kriss, J. and van Ham, F., Talk Before You Type: Coordination in Wikipedia. *Proceedings of HICSS-40*, 2007.
35. Wattenberg, M., F. Viégas, and K. Hollenbach, Visualizing Activity on Wikipedia with Chromograms, *Proceedings of Human-Computer Interaction – INTERACT 2007*.