# Interactional Identity:
# A Conceptual Tool Designers and Developers use to make Joint Work Meaningful and Effective

**Judith Brown**
Psychology Dept.
Carleton University
mmjbrown@gmail.com

**Gitte Lindgaard**
Psychology Dept.
Carleton University
Gitte_Lindgaard@carleton.ca

**Robert Biddle**
Computer Science
Carleton University
Robert_Biddle@carleton.ca

## ABSTRACT

We studied collaborating interface designers and software developers engaged in multidisciplinary software creation work. We interviewed 21 designers and developers in 8 organizations to understand how each specialist viewed team interactions. We also shadowed most of these participants. A grounded theory analysis of interview transripts showed that designers and developers construct unique identities in the process of collaborating that provides meaning to their real and virtual, artefact-mediated interactions, and that helps them to accomplish the work of creating novel software. Our model of interactional identities specifies a number of aspects of joint project work in which an interactional identity is expressed. We suggest these identities are constructed to bridge a gap between how designers and developers were taught to enact their roles and the demands of project work. We further suggest that such identities may be common in other types of highly collaborative and creative work. Knowing that such identities exist can help individuals to consciously select and develop more meaningful and effective identities. It may also help project managers or team members understand the behavior of other team members.

## Author Keywords

Collaboration, personal identity, self, interaction identity, design, development, software

## ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous

## INTRODUCTION

We see collaboration as purposeful, joint action that includes a significant amount of communication between collaborators. Collaborative work creates shared understanding, which leads to the development of shared goals and visions of a group's desired outcomes. It can also serve to make concrete progress toward a group's desired outcomes. It is easy to see why collaboration would be an important part of many complex activities since they often cannot be accomplished by a single individual.

Software creation work is a complex, creative, artefact-rich, multidisciplinary, and highly collaborative activity. When we began our research we wanted to understand what meaning shared design artefacts had for software team members as they engaged in the early stages of software creation work. This open-ended and influential stage of the work lays the foundation for the creation of real interactive software and is sometimes called the concept design stage. To focus our research we studied interaction designers and developers because they held key roles on the team and because their work overlapped significantly. Our aim was to understand the collaborative work of designers and developers in depth in order to contribute to the development of better tools that would support their collaborations.

In the course of this research we discovered that software designers and developers construct project-specific identities that appears to 1) imbue meaning to their talk- and artefact-mediated interactions with other team members and 2) provide direction to their team interactions in order to make them effective. This paper argues for the existence of, and also explicates, this pervasive and valuable conceptual tool that we have named 'interactional identity' and that appears to be an essential, pervasive, and largely unexplored aspect of collaborative work. We explore many aspects of this identity, including how it is tied to artefact use, in this paper.

### Motivation

Understanding collaborative work is an active research area that crosses many domains. For example, collaboration has been studied to meet business aims, to support and sustain community organizations, and to ensure positive health outcomes in patients. We investigate collaboration from the perspective of collaborators. And, more specifically, we study it to contribute to the large body of research on the creation of better software products. However, because our research investigates a conceptual, cognitive tool that collaborators in general would likely use, our research is of potential relevance to multiple domains.

This work provides evidence for the existence of interactional identities. A large body of research exists on identities
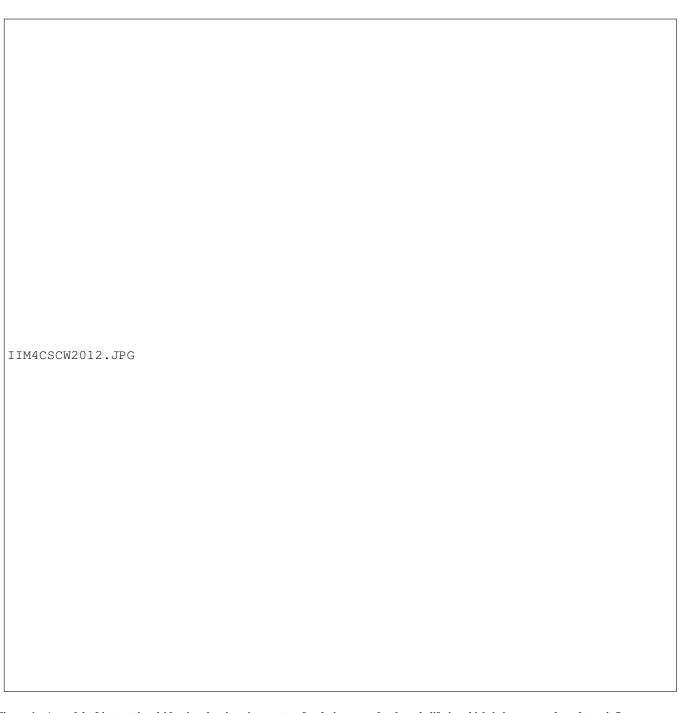
IIM4CSCW2012.JPG

**Figure 1. A model of interactional identity showing six aspects of a designer or developer's life in which it is expressed, and two influences on its specific incarnations for an individual: previous experiences and current circumstances and desired outcomes. The 6 categories were 6 of 21 categories that were the outcome of the open coding step. The higher-level category "interactional identity" was the outcome of the next step, axial coding.**

because identities are an aid to understanding an individual's actions, thoughts and even feelings [?]. We focus on interactional identities at work, seeing them as one element of work identities, which are an important aspect of the overall identity of many adults. We show that such identities are meaningful to those who construct them because they help that individual to make sense of their team interactions. As such, we hypothesize that these constructs satisfy a personal need for a sense of self and a sense of self in relation to others. Similarly, we will show that these identities are also used by individuals to structure and direct their actions in the world. In this sense, we hypothesize that interactional identities influence the outcomes of collaborative work.

**Related Studies in Software Work and Identities**
Some researchers have studied relations between identity constructs meaningful to software workers. Fidel and Garner asked which identities were strongest in computer professionals when they surveyed 125 computer professionals with master's degrees. Focusing on the impacts of organizational identity and professional identity on personal identity they found that "Computer workers with advanced degrees have some elements of a professional identity, but show little evidence that this identity conflicts with or supersedes their identity as corporate employees" ([?], p. 122). Carmel focused more on the influences on developers' professional identities at work. He studied developers in six countries and investigated how national identities contributed to the professional identities of software developers in each of the countries [?]. He found that professional identities of software developers are influenced by national myths. Collectively, this line of research portrays software workers as individuals who are actively engaged in selecting or constructing appropriate shared cultural identities.

Other work, more similar to ours, addressed enacted identities. Of relevance to software engineers, Brown studied engineering identities in the workplace and showed how these are deliberately influenced by employer actions and actively shaped by engineers in response to many factors in their professional and personal lives [?]. Kilker studied 26 student software teams and found that team members enact either technology-centric or socially-centric "design identities" [?] when interacting with each other. And, Turkle and Papert contrasted the largely negative American cultural identity of "those who compute" with the practices of computer programmers and found individuals created their own personalized "style of working", which reflected concrete "ways of knowing" [?]. Their work indicates that cultural identities can be rejected and suggests that personalized identities can be constructed by programmers in their stead. Collectively, this line of research suggests that enacted identities, which are shaped by many factors, are the most relevant identity in workplace settings.

Our work is grounded in a theoretical perspective (a socio-cultural approach called activity theory), uses rigorous methods for data collection and analysis [?, ?], and focuses on identities constructed in the context of real project work in organizational settings. In a socio-cultural or activity-theoretical framework [?, ?, ?, ?, ?] identities in general are theorized as cognitive tools that can be used within a motivated activity system. Identities are also often seen as being constructed in the process of engaging in an activity [?], or as an outcome of an activity, as in van der Riet's work where she concluded sexual identity was an outcome of engagement in community activity [?]. It is also not unusual for work on identity from an activity theoretic perspective to make links with artefact. For example, Vagan suggested "A sociocultural perspective is needed to capture how the incorporation of artifacts [sic] provides people with tools of agency and identity; how artifacts mediate, expand, and limit action; and how they work as tools for individual's identities in cultural worlds." ([?], p. 45).

Our work is most like Martin's research where she identified a number of customer roles in agile software teams by interviewing customers. Her customer "direction setting roles" and "collaboration guides" do at time seem very much like some of the interactional identities we have identified, but she has interpreted these as different aspects of the "customer" role distributed across a subset of the agile team [?].

**Method**
Twenty-one designers and developers on eight multidisciplinary software teams in 8 organizations volunteered for this research. The teams were in the early, open-ended stage of a particular concrete software project, which required significant amounts of team interaction. This occured mostly face-to-face, since the teams were not distributed. Many factors varied between the teams including the products they were designing, processes, organizational resources, and their market. The role of the designers on the team varied greatly from a graphic designer with responsibility for screen design to a user experience designer with much broader responsibilities. Their years of experience in the discipline ranged from 2 to 40 years. The developers were mostly university computer science graduates, although one was a trade school graduate and another was self-taught. Their years of experience in the discipline ranged from 3 to 22 years.

We conducted 21 hour-long interviews with designers and developers who were part of these 8 teams and observed 14 (67%) of these individuals at work. When observing within organisations we first followed the designer for several days and then the developer. We purposefully chose days where the project work was the most important priority for the designer and developer. At all organizations we took notes and photographs of artefacts since these were prominent in interactions. These artefacts were hand- or computer-generated. Most meetings were face-to-face, but email was a common backup communication mechanism. The interviews were the primary source of data for this analysis; other data served to confirm and enhance understanding.

The method used to analyze the data was grounded theory [?]. Grounded theory is a method that allows researchers to analyze large amounts of integrated data from many different sources to identify patterns in the data and produce theory. At the heart of the grounded theory method is the method

of constant comparison. Analysts create carefully defined codes and then higher-level categories that they constantly refine, combine, or break up as they test emerging theories about the data. The method keeps the analyst close to the data and therefore the categories or theories that emerge are a true reflection of the analyst's approach and the data that was collected. This analysis introduced a variation to the standard use of grounded theory in that it was used to generate a theory about individuals rather than groups. Throughout, Atlas.ti was used to aid with coding, comparing, counting, and finding data. More than a thousand quotes were analyzed in this way.

## DERIVATION OF THE IDENTITY CONCEPT

The identification of the concept *interactional identity* was the major outcome of the grounded theory analysis. The first step, the open coding step, a systematic perusal of an initial set of transcripts, produced 15 saturated categories. As a simple example of an action in this step, a quote such as "My goal for this project is to ensure that my fellow team members converge on a design concept quickly, but remain flexible to making changes to it." would suggest the creation of a category 'Personal Goal'. As we found other quotes expressing a participant's personal goals for the project (from the same participant and others), we would code these similarly and after a while the category 'Personal Goal' became saturated and the category served as a tag or container to collect all of these similar quotes together. Fifteen other saturated categories were produced in this manner. The next step was axial coding. We produced a large spreadsheet to reflect on the quotes for each participant across all the categories. This showed that for the initial set of participants 8 of the 15 categories were related to each other through a higher-level category we labeled "interactional identity" because the quotes in each of these 8 categories reflected an individual's interactional identity i.e, they each provided information about how the individual viewed collaborative work. This new unifying theoretical concept then became the central interest of the analysis. As a consequence, 7 categories that were of no particular relevance to this concept were dropped. We present six of the eight categories here, dropping two more that were insufficiently distinctive. Figure 1 shows the six categories and the one higher-level category in relation to the others.

The next sections contains two detailed case studies: developer Wil's and designer Owen's. We show how the higher-level category interactional identity was derived from data that was coded with the six categories shown in Figure 1. We show how we identified "interactional identity" as a distinct theoretical construct, and also how we identified Wil and Owen's specific interactional identity. Following this, I briefly list 19 other interactional identities we derived to provide further evidence for the existence and nature of this construct. We then discuss how this construct relates to theoretical work on identity in general. To conclude we examine the specific interactional identities that emerged from the analyses and the insight these provide regarding software creation work. These insights came from asking why designers and developers create interactional identities generally, and why did they create the specific interactional identities we observed.

## DEVELOPER WIL'S INTERACTIONAL IDENTITY

Wil was one of three developers on a project team and was responsible for implementing the product, supervising a junior programmer, and liaising with a third programmer who was designing a database for the back end. He had 22 years of experience as a developer. In interactions with others on his team he saw himself as a "binder" or someone whose actions cause two elements to cohere or stick together in a mass. In Wil's case the two elements to be bound were a variety of different artefacts (e.g., a mock-up of the user interface, a story of user interaction, or a database scheme) and the glue that caused the two elements to stick together was the program code he wrote and rigorously tested using a suite of test cases he also coded. In effect, by 'binding' Wil produced the software.

The following subsections outline key pieces of evidence (i.e. quotes made by Wil) coded using each of the six concepts that supports the claims that 1) an interactional identity exits and that it is reflected in the concepts identified in the model, and 2) that developer Wil's interactional identity is "binder".

### Wil's Personal Goals for the Team's Project

Wil's personal goal was to create the required functionality and test that it works as it should. This is a large and consuming task.

> Interviewer: How do you know when your goal in the project is achieved?
> Wil: . . . when I can execute the app[lication]. . . We write tests . . . to test . . . each unit of work—like an object or a data access—so [that] . . . at a certain level we know that it works—-the functionality's been achieved.

### Wil's view of the Team's Shared Object/ive

In Wil's view, each person on the team is responsible for certain aspects of the project and producing certain artefacts. For example, the designer is responsible for addressing business issues, client issues, and specifying the look, feel, and functionality of the user interface as well as creating user stories. Wil sees the designer as a buffer between himself and the business and client's worlds. Wil says the interaction designer "gathers the business requirements from meetings with the client and [also the] documentation from them. He gathers the requirements and fleshes that out into user stories." Wil is happy to *not* be the person interacting directly with the client and happy to have the interaction designer solely responsible for determining the user interface and producing artefacts that he can use.

### Wil's View of the Team's Shared Artefacts

For Wil, artefacts play an important role. Single shared artefact represents a kind of high-level map that indicate the "path" someone else wants to take with their aspect of the work. These maps can indicate what a client or interaction

designer wants in terms of the product. These artefacts play a strong role in guiding what Wil codes.

In Wil's view, new artefacts are created from others. Wil is sensitive to how certain artefacts receive client approval (e.g., the stories and mock-ups), revealing his sensitivity to the stability of an artefact and his need to have stable artefacts. Some of his views are revealed in his generalized description of the design process in his workplace:

> The main stakeholder is the client. They ...provide the ideas and their vision of what they want the application to do and what it looks like. ...The ID [Interaction Designer]— gathers the business requirements from meetings with the client and documentation .... He gathers the requirements and fleshes that out into user stories. From that he would build mock-ups representing those user stories. Those would be approved by the client. And once that's done, myself [the developer] and database analyst would ...start looking at the [software] design overall, [the] high-level architecture [for the code] [To produce something that would show] what the application [meaning the software's design] would look like, and the database schema [would look like] on the database side.

In the following quote, Wil discusses mock-ups, stories, and database schemas in relation to the code that he must write.

> Wil: During the estimating of the effort, [The designer] would do a walkthrough of the mock-ups [guided by the stories] so that we [the developers] understand what's required. ...From that meeting we get a lot of knowledge to help us in designing it [the structure of the software]. ...And [the database analyst] ...he's at those meetings as well, to get an understanding.
> Interviewer: He seems to always have his database schema in front of him and is always considering the implications to that schema.
> Wil: Right. Exactly. Cause he'll use the mock-ups, as well [as the stories], to come up with the schema. And ...on the developer's side, we do the same thing ...[but using] his schema and the user story and we try to create the [software] objects that'll *bind* those two together.

As Wil sees it, new artefacts are created from old ones by "binding," an unusual verb Wil used that caught our attention. For instance, Wil saw stories and mock-ups as artefacts that he would bind through the software objects he created. Wil's binding action produces software, and in Wil's view, sequences of binding actions (by himself and others, whom Wil also sees as binders) complete the application.

### Wil's view of Tensions in the Team's Project Work

Wil's aim was to protect himself from project tensions. Tensions arose in Wil's work when artefacts changed, because this created a tension between what others required and what he had coded. Wil wanted to remain productive and to focus on the work of binding. He appeared to find that the project moved forward when team members agreed on the key artefacts (e.g., stories and mock-ups).

> Wil: ...sometimes, a picture is ...far easier for people to understand, [it] makes it clearer. It's always at a high level-right?—[the] relationship between objects, or relationships between pages—like navigation, ...or what's gonna be displayed on the page. Things like that. ...sometimes you wanna just ..., brainstorm through it to see if it makes sense. And then, once you've ...decided on a particular path then, ...everybody lines up on the same page.

Wil protected himself from wasted effort by coding the more stable parts of the application first. For instance, he understood that the mock-up was volatile and the stories were more stable, so he relied more on stories and coded the stories (i.e., the behavioural aspects of the user interface) before coding the visual aspects of the user interface (i.e., the mock-up).

### The "Binder" Identity as Conceptual Tool

Wil's case was like all others analyzed in that all of the coded categories together indicated one singular identity. Wil's goal was to make sure the application worked. He saw shared artefacts as things that team members produced (because each team member had responsibilities for different aspects of the the user interface) and things that he used in his "binding" work. The "binding work" produced tested software. Tensions arose within the team when shared artefacts were not stable. In particular, for Wil unstable artefacts meant that he would be presented with barriers to achieving his personal goal of ensuring the application worked.

Wil's case was unique in that no other developer presented with this particular interactional identity. The "binder" identity served as a cognitive tool to help Wil respond to unique project circumstances, collaborate, and achieve shared project goals. Wil's project was in a constant state of flux. No one, not even the clients, could clearly specify the requirements, since these were constantly evolving. Part of the envisioned application may not have even been feasible, and on top of this, the product was threatening to some large businesses who did not want to see it completed. Wil's identity as binder provided an additional meaning to his interactions. It meant that he could shield himself from much of the project's turmoil. It allowed him to focus on producing quality code. He looked at artefacts in terms of their potential to enter into his binding work, and he looked at producers of artefacts as agents in that process. This was a positive way of making sense of his work and likely also provided him with motivation and a sense of well-being in the challenging circumstances he found himself in. Wil's identity also helped him to be effective because it gave him a way of assessing the work of others in the workplace, and helped him to prioritize which artefacts he should rely on first for his work (i.e. stable artefacts). Wil's identity arose out of his previous experiences (in particular knowledge of binding processes in another context and 22 years of experience as a developer), the current project circumstances, and his genuine desire to contribute to the success of the project.

### DESIGNER OWEN'S INTERACTIONAL IDENTITY

| Designer handles | Descriptive phrase | Yrs. exp. |
|---|---|---|
| Incorporator | Incorporates their design work with the work of others. | 2 |
| Pioneer | Opens up new work in the field of user needs analysis. | 2 |
| Negotiator | Negotiates with others to arrive at design decisions. | 6 |
| Closer | Closes in on a design deal that is agreeable to all stakeholders, including himself. | 8 |
| Reformer | Reforms organizational structure to achieve greater compliance to design standards. | 10 |
| Movie director | Acts as a director of a cast and crew in rehearsal mode for production of a movie. | 10.5 |
| Quality advocate | Advocates for product quality. | 11 |
| Steward | Acts as the teams steward for producing useful software for his products target audience. | 12 |
| Facilitator | Acts as a facilitator of a distributed, problem-solving activity. | 13 |
| Teacher | Engages others in courseware production through teaching. | 33 |
| Artist-in-business | Uses his art to achieve business goals for the team. | 40 |
| Elder | Sees software-creation work evolving over time. | 40+ |

**Figure 2. Designers' Interactional Identities—Handles, Descriptive Phrases and Years of experience**

| Developer handles | Descriptive phrase | Yrs. exp. |
|---|---|---|
| Assembler/builder | Assembles parts from others and produces parts of his own to build a product. | 3 |
| Catalyst | Acts as a catalyst in the team to ensure changes in the product meet the team's objectives (e.g. usability). | 7 |
| Problem solver | Seeks satisfactory solutions to perplexing and difficult software problems. | 10 |
| Practitioner | Reliably and professionally delivers and expects a high standard of work in the team. | 10 |
| Historical anomaly | Holds a unique position in software work and is misunderstood. Must constantly explain role to others. | 15 |
| Training advocate | Advocates training concepts and uses training techniques to deliver this message to the team. | 17 |
| Enabler | Provides other team members with adequate means, opportunity, or authority to do their work. | 19 |
| Pragmatist | Aims to help the team establish and meet achievable software goals. | 20 |
| Binder | Aims to develop software that binds together the work of others to create something that is functional. | 22 |

**Figure 3. Developers' Interactional Identities—Handles, Descriptive Phrases and Years of experience**

Owen was a lead game designer and chief technology officer in his organization. He had 10 years of experience in design work. He was one of five people creating a game for an established client. Like Wil's interactional identity, Owen's interactional identity of movie director "shines through" in the aspects of Owen's work identititifed in the model depicted in Figure 1. However, unlike Wil, who like most other participants, was not conscious of his interactional identity, Owen represents the extreme case of someone who was more self-aware in this respect.

**Owen's Personal Goals for the Team's Project**
One of Owen's goals in his work as lead game designer was to find "the central motif" for the game. As Owen said, "The main thing in design is the concept, or the motif, and at this point [the very early stages of design] I have no idea what that's going to be". Another of Owen's goals was to motivate people so that they cared about their work. He also encouraged the team to explore and be creative. To do this, on occasion, he would deliberately make himself look foolish to allow others to take risks.

> You often have to, um, give . . . help them [the team] . . . break out of their shell by doing it yourself first. . . . like that example I gave, "Do it really, really loud, and then do it really, really soft." If I do that, and I take the risk of looking

foolish, then you [someone who sees Owen taking risks] are more comfortable [taking risks]. If I don't think I'm getting enough [risk taking], if I come out and I take it really, really loud and really, really big and make a real fool out of myself in front of everyone, it's easier for the actor to do that. I found that worked really, really well, working even on the design [of the game].

Owen was unique in his description of making progress in design work through the initiation of exercises to promote risk-taking and creativity. This and Owen's talk of motifs evoked the creative arts.

**Owen's View of the Team's Shared Object/ive**
The team's shared object was the game's user interface that they would jointly create. Owen believed that as a lead game designer he must hold an opinion about everything, else people seeking feedback would get frustrated. Consequently, he relinquished control over parts of the game, but "offered opinions", because if you don't offer an opinion, "they get frustrated". "It's [offering an opinion about everything is] very similar to what a movie director has to do". "I learned

this, actually, believe it or not, when I was in film school". Using his past experience with film and an analogy with movie directing, Owen explained his interactions with the team as follows:

> The director has to have an opinion. If the director comes in and says, "Oh, just give me something groovy."—an expression I learned from my film school days—that's what you don't wanna do [is give vague directions]. ... [Instead, what's required is to give specific directions, such as] "just move the cam[era]" —they [directors] have to have an opinion about everything. And that's why they're called a director. [They] direct every little thing. But they can't do anything. The same thing seems to happen in game design.

Owen's view of the team's shared object/ive (the completed game) was that each person had a role, but that he, as a director, offered opinions about everything. Owen was very clear about this role.

> I've seen it [the design process] in film, as a director, and [as a hardware designer] designing [chips]—you know, 'cause that [creating a film or creating a chip] is just a design problem. I've seen it in hardware, designing laser chips. And now I'm seeing it in [designing] games—software. I've seen it in three different things, and it's absolutely fascinating 'cause everything I've learned abstractly applies. It's the same. I take things I've learned from [film] ..., and hardware, and they're applicable here as well.

In Owen's view, the team had a shared objective (creating the game), but he played a strong role in helping individuals "play their part" by providing strong opinions about their work. Owen used an analogy to the work processes of hardware designers and film directors to describe his team interactions (the reasons for providing strong opinions) and this analogy provided a strong clue (but not definitive clue) as to his interactional identity. Although he claims his design experiences in film and hardware were influential, we checked to see if one identity or the other "shone through" more strongly in all of the categories in the interactional identity model and in his behavior at work.

### Owen's View of the Team's Shared Artefacts

Owen sometimes sees artefacts as props in a drama. In one story he told us he literally organized a game re-enactment in the office with the designer, writer, and developer to explore variations for the "voice" of the gaming characters. This worked particularly well because the setting of the game was a virtual office. In the description below, the props the team used were a representation of the gaming characters created by the artist, and the reenactment in Owen's office space was a reenactment of the game itself, which Owen considered to be a "prototype concept". Owen saw the shared artefacts as valuable because they encouraged fun and exploration, which helped with problem solving.

> So we [the gaming team] put together a simple proto-

type concept—just take the [gaming] characters around the [real] office, and do some silly event around them. ...We had me and [the developer] and [the writer] and [the graphic designer]. They [all the characters in the game] were all in there [in the reenactment]. [The graphic designer] just drew them and—it was fun. Um, and then I told [the writer], I said, just try different voices, ways of doing each of the things for each of the people. You know, don't worry about being consistent; just expand [explore different voices]. ...That let her try 6 or 7 different things [voices for characters], ..., and then we could play them [act the game out in the real office setting]. Then after playing them she said, "This one's really working well for me," and we converged on that, we picked that [voice].

Owen sees artefacts as props in a production and elements around which drama unfolds under his leadership.

### Owen's View of Tensions in the Project Work

Owen sees tensions within the team as creative opportunities. For example, Owen described a tension between the contents of his game design document, which contained the game's mechanics, and the team's understanding of it. His view was that resolving this tensions resulted in the creation of a 'motif' (i.e., the central idea of the game), which, once identified, would serve to motivate the team to continue their work.

> We now get this tension, because for some reason they feel that what I wrote down in my first draft [of the design document] is ...cold or brilliant. ...And they don't get it. And because they don't get it, they struggle and try to ...understand it. And they're not [understanding it] 'cause there's a lot of problems with it. ...Through iteration [of the game] and through discussion [with the team], my ideas start to clarify. And at some point, what magically emerges is this concept, or this motif, that everyone suddenly grabs onto and gets. And it becomes this unifying aspect—...the method that everyone rallies behind.

He also described the work of creating games as a dramatic process that has a built-in tension between the forces of expansion and contraction.

> Initially everything expands, expands, expands, expands, expands, as new ideas get thrown in the mix. And I keep trying to incorporate those ideas. I know ...I'm on the right track when I start taking things away. As soon as my mental framework moves from adding to taking things away, I know it's near[ly] finished.

At other times, Owen discussed design work in terms of cycles of abstraction and concretization. "There's holes, it's horrible. And at this point what I try to do, is we try to start reducing it down to something concrete again. So I started concrete, went abstract, and then go down to concrete".

Owen's theory about design in general is that it is inherently imbued with tension and drama, but that this is a good thing because when the team works through these, it results in the

creation of a game.

### The "Movie Director" Identity as Conceptual Tool

Owen's case was like all others analyzed in that all of the coded categories together indicated one singular and unique identity. To summarize, Owen's personal goal was to find the central motif for the game, which would energize the work of the team. He contributed most towards the advancement of the team's end objective by keeping the team motivated, encouraging creativity, and providing opinions about everything. He saw shared artefacts as things that the team used in dramatic re-enactments of the game play or elements around which the drama of game design unfolds. Tensions were inherent in game design and seen positively as part of a natural and dramatic process that produces a game.

Owen's "movie director" identity served as a cognitive tool to help Owen respond to unique project circumstances, collaborate, and achieve shared project goals. Successfully completing Owen's project (a learning game) required a significant amount of creativity from all the team members and the coordination of the work outputs of 6 different types of designers and one developer. From the perspective of this challenge, it made sense that Owen would see himself as a "movie director", since no one else on the team had particular responsibility for keeping the team motivated and keeping the creative output high. This was an identity he crafted for himself, because it drew on his skillset, filled a need of the team, and because he was very invested in the success of the project. It provided additional meaning to his interactions, made his work more interesting and pleasurable, and contributed to his sense of self-worth (because movie directors play a central and critical role). Owen's identity was also effective because by using it he could draw on motivational techniques for the team and also techniques for augmenting creativity. In this way he produced better games more effectively. Owen's identity was a response to the current project circumstances, his previous experiences, and his strong desire for the project to succeed.

Using grounded theory we discovered that an individual's interactional identity was reflected in the talk in interviews about an individual's personal goals for the project, their view of the team's shared object and artefacts, and their view of tensions on the project. Owen and Wil's specific interactional identities were derived by examining data, i.e. their quotes on these four topics. In both cases, the specific interactional identity captured the unique way that Owen or Wil brought meaning to their work (their purpose), their view of the role of shared artefacts, and interactions with project team members, and their view of project tensions. We observed Owen and Wil at work and in both cases, the interactional identities were confirmed by observing behavior. Owen worked with others, as he understood a movie director would interact with his cast and crew (without, incidentally, exposing his identity overtly). Wil's interactions with others on the team also corroborated his interactional identity as binder—he was particularly attentive to artefacts, was disturbed when they were unstable, and did write application and test code while referring to multiple artefacts.

## OTHER INTERACTIONAL IDENTITIES

Nineteen other interactional identities were identified. The overall process for identification was always the same. After recruitment, if possible we would shadow a participant in their workplace for a few days. Then we would interview a participant and ask them about the particular software project they were working on. We would then ask them questions about that project, the team, and the process being used. We asked questions about the artefacts they used to support communication within the team, the end product, and about their interactions with their fellow designer (or developer). We also asked about project tensions and their resolution.

We then transcribed and coded comments— especially those that had to do with a participant's personal goals, their view of the team's shared object (the user interface), their view of the team's shared artefacts, and lastly their view of tensions in the project work. Looking at the quotes in each of these four categories and sub-categories in turn we first discovered that these categories provided clues about who a person was in action. After reaching saturation, meaning once we were sure of this relationship, we reanalyzed the data to try to discern a handle that would reflect what a participant said, usually "between the lines", about who they were when interacting with others on the team. We particularly paid attention to unusual verbs (e.g. bind) or analogies (e.g. directing), that could be useful. The main goal was to find a handle that fit with the quotes in all four categories. As we examined an individual's data we would try and and then discard handles until we found one that was completely satisfactory in the sense that it fit all the relevant data and our observations. For instance, for the enabler, whose transcript we analyzed while writing this paper, we explored three other handles (fitter, scheduler, deblocker), as we processed the data, before arriving at one (enabler) that fit all the data perfectly. All of the details for all the other analyses are publicly available **reference to dissertation here** and we touch on a few more cases below.

Massie, is a designer in charge of developing a web course. Massie was a relatively new designer and was working with a very experienced developer who was also interested in designing courses and not just developing them. How could she make this delicate situation work? While she welcomes the developers input and values his experience, which is significantly greater than her own, she stops short of allowing this developer to take over the design work. By assuming the identity of a classroom teacher, she encourages the developer to contribute, but still controls the developer's contribution. Below, in describing how she would interact with this developer she uses the Socratic method, a technique commonly used by teachers. Massie, like a good classroom teacher, remains in control.

> Interviewer: Do you ever find that there's a grey area where your roles overlap and he [the developer] might do some of what you do and you might do some of what he

does, traditionally.

Massie: Yeah, because he's [the developer's] really interested in design too. And you can't not. You know, . . . some people in design are really interested in development too.

Interviewer: Is the developer's interest in design OK with you?

Massie: Yeah, it's OK with me. . . . I suppose it could happen that you'd have to tell somebody [meaning a developer] to back off . . . taking over. But like, they [the developers] don't actually have this upfront piece with learners [normally they are not involved in the initial stages of course design which involves designers working with the clients and end users], but I invite him [this particular developer] to participate [in these client meetings] because he's interested. . . . I mean, if you did have a developer who was saying, No, I really think you should go in this direction with the course, [and who was at odds with what was going on in designer-client meetings] then, you know, you'd have to have a meeting [to resolve this tension]. But . . . when somebody [meaning this particular developer] is that interested, . . . you'd probably say where we're [the designer and client are] trying to go [with the design], . . . and from discussions with the client [you'd say to this developer] [We know] we wanna go here—how can you help us get there? How? And You're [this particular developer] interested in learning or you've studied a lot yourself. If you were that particular learner, what would you think?

Our focus on concrete team projects, rather than design or development work in general was fortunate because interactional identities are tied to a particular activity.

Also, because we found interactional identities are normally non-conscious identities we did not ask directly about the issues we were interested in, but instead asked many project-specific questions, and during analysis focused largely on the data that we found was reflective of an interactional identity. These data revealed how that participant saw themselves in interaction with other team members. It also helped to provide additional meaning to our observations in that the interactional identity proved to be very good at explaining observed behavior, allowing us to develop an understanding of *why* the work unfolded in a certain way.

Not all data revealed an individual's interactional identity. For example, the data that we coded

We found that each participant had a unique identity and these are shown in Table 3.