(META)MODELING “CONSTITUTIVE COMMUNICATION”: TOWARD A REAL-TIME REFLEXIVE INFRASTRUCTURE FOR COORDINATION AND CODESIGN

By

Nathaniel Joseph Savery

B.A., University of Connecticut, 2005

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirement for the degree of

Master of Arts

Department of Communication

2011
This thesis entitled:

(Meta)Modeling “Constitutive Communication”: Toward a Real-Time Reflexive Infrastructure for Coordination and Codesign

written by Nathaniel Joseph Savery

has been approved for the Department of Communication

___________________________________
(Dr. Michele H. Jackson, Advisor)

___________________________________
(Dr. John M. Ackerman)

___________________________________
(Dr. Leysia A. Palen)

Date __________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.
This thesis explores the complex relationship between communication and design practice as they occur within the development of technologies. Contemporary theories of communication and technology provide a conceptual basis for treating communication and design as reflexive structuring acts that change the affordances of an interaction situation while embedded within an environment that constitutes the situation. Drawing upon some of these theories, a design research project is undertaken to define a general infrastructure for real-time interaction that affords users reflexive capabilities for redesigning and restructuring the relational situation from within. The design solution developed here proposes a variety of strategies to model the emergence, complexity, and multiplicity of objects as the negotiated outcomes of situated human-computer interactions.

In order to consider the feasibility of this design, an inquiry is performed to assess contemporary approaches to reflexive infrastructure for real-time interaction. Various existing collaboration and coordination frameworks and support environments are examined that articulate solutions to elements of the problem space outlined in this thesis. The analysis focuses on the place-based and activity-based approaches to representing dynamic interaction situations exemplified by the research systems Orbit and Intermezzo. The way that these approaches enable and constrain the development of dynamic interaction situations provides a ground for
considering the feasibility of the proposed mechanisms as means for reflexively modeling responsive emergence. The design research project undertaken here results in a more concrete proposal for design of infrastructures that reflexively model complex relationality and support emergent forms of coordination and codesign.
Acknowledgments

I am blessed with many, many wonderful people in my life who have made this project possible through their inspiration, support, advice, and friendship. On the academic side, I would like to thank my committee members, I am very grateful to each of you for your influence on my scholarship and for the myriad ways you’ve supported my development as a person and scholar. In particular, I would like to thank Dr. Michele Jackson for serving as the chair of my committee and for challenging me throughout the process to develop my thinking and improve my scholarship. I’d also like to thank Dr. Leysia Palen and Dr. John Ackerman for serving on my committee and providing invaluable advice, insight, and support along the way. I am also grateful for the intellectual stimulation and friendship of many graduate students, faculty and staff from the CU Department of Communication. On the personal side, this project would not be possible without the patience, love and support of many wonderful people. I want to recognize especially my family and close friends for their love and support throughout this process and throughout my life. Thank you to my mother, Paige and also my father, Jay, for always believing in me, sacrificing for me, and guiding me to be a good person. Also, thank you to my sister Caroline, for being a caring friend, creative inspiration, and shining example of courage in action on a journey parallel to my own. Thank you to my closest individual friends over the years; I am very grateful for your love and support. Also to my larger groups of very good friends in Boulder, CO, in Salt Lake City, UT, and in central CT where I grew up and spent most of my life—if not for the ways in which each of you have nurtured, inspired, and challenged me to discover my own path, the current work would not be possible. Thank you all for sharing your journey with me and for helping me to find my own way.
The articulation of the “Hmmm…” system that I am putting forward here is an extension upon years of codesign involving numerous people, conversations, texts, and experiences. While I claim the overarching vision as my own and take full responsibility for any and all shortcomings, some of the particular ideas and proposed functionality contained herein were shaped by family, friends, scholars, artists, technologists, activists, and occasionally strangers over the years. In particular, the evolution of this vision into relational technologies has been influenced over the years in conversations and design sessions with one of my good friends whom I’d like to acknowledge as a codesigner, Dr. Leonardo Palacios. Thanks Leo, I am very grateful for your friendship and willingness to “jam” with me over the course of years with the various conceptual iterations of what has become the “Hmmm…” system. Furthermore, I want to say thank you to every person over the years who has been willing to genuinely listen and respond to me and share their experiences, and to everyone for just being patient with me in general.
## Contents

Chapter One ........................................................................................................................ 1

   Introduction ..................................................................................................................... 1

Design and Communication ............................................................................................ 3

   Design as a communication issue. .............................................................................. 7

   Design-in-communication ......................................................................................... 7

   Communication-in-design ......................................................................................... 9

STS and the Relational Boundaries of Design ................................................................ 10

   Design as negotiation and stabilization .................................................................. 11

   Infrastructure and background. ................................................................................. 13

   Communication as embedded design ...................................................................... 16

Design in Human-Computer Interaction ....................................................................... 18

   Situated action and technology design. ................................................................ 19

   Second and third wave tensions ............................................................................... 24

   Continuous redesign ................................................................................................. 26

Reflexively Theorizing Within a Metamodel ............................................................... 28

   Communication as constitutive metamodel ............................................................. 30

   Extending a metamodel to technology design ......................................................... 33

Conclusion .................................................................................................................... 36

viii
Chapter One

Introduction

Communication is much more than transmitting information—it is also how we coordinate our shared activities and negotiate our cohabitation in a shared world. Modern communication technology changes the practices of human communication, creating new ways for us to connect and coordinate with one another. Yet new technology has deep effects on our environment as well, creating reliance upon new infrastructure as technology becomes a commonly accepted and even invisible part of life in a society. Technology changes the outcomes of communication, in ways we often don’t see, as it becomes part of the background and is taken for granted as a factor in the communication process.

Humans design technologies. In the development of a technology, designers and users construct meaning about the technology and its relationship to the world we share. These interactions shape the design and use of technologies, clearing the way for certain interpretations and social possibilities while closing off others. Here, communication plays a major role in the outcomes of design, which, in the form of technologies, simultaneously play a major role in the outcomes of communication. There is a reflexive relationship between communication and technology and their effects in everyday life and experience. We can see this reflexive relationship playing out in the design of communication technology intended to change communication possibilities in the world. Here we see communication and design reflexively influencing each other; designers construct artifacts that are outcomes of communication about communication. At the same time, these artifacts are factors that change the rules of communication and influence future possibilities for meaning-making. The artifacts that result
from design arise *from within* communication processes, but they are also new participants in the shared world of communication and design.

A complex relationship exists between communication, design, and technology. For a designer trying to build not just communication technology, but technology that could change the possibilities of design itself, this complexity becomes a central issue. Could designers build a technology that deliberately incorporates the reflexive role of communication in the process of design? That is the question that I seek to address in this thesis, considering how a theoretical understanding of relational structure achieved in communication might inform the design of a metatechnology that models a complex relationship between communication and design. The resulting metatechnology would seek to model the emergence of structure as it occurs and evolves within ongoing communication and design processes.

In this chapter, I aim to unpack the complex relationship between design and communication. I begin by defining these terms and showing how, in practice, the two are interconnected in various ways. Questions raised about the entanglement of social and material dimensions in practice provide a way to think about communication and design as processes that condition, and are conditioned by, our embodied experience of being in the world. Work within the field of human-computer interaction has taken up this complex relationship from a designer’s perspective in attempting to understand and build technologies that consider the reflexivity of communication and design in various ways. Like human-computer interaction studies, scholarship in communication and technology studies also share an interest in understanding this reflexive connection that theorizing and making sense of the world has to the experience of being in, and designing, the world we live in.
Each of the many ways of theorizing about the world leads to different practical possibilities and outcomes for communication and design. I argue that the idea of a metamodel as a conceptual framework based on the reflexive relationship between communication and design in the world holds promise for connecting the various practical and theoretical forms of design and communication and allowing them to be in productive conversation with one another. To explore this possibility, this thesis attempts to extend the notion of a conceptual metamodel to the prototypical design of an actual technological artifact that provides an interactive, evolving model of the complex relationship between communication and design.

**Design and Communication**

“Everyone designs who devises courses of action aimed at changing existing situations into preferred ones“ (Simon, 1969, p. 129). Following this broad definition, design can be considered a practice and an orientation to the world. Many contemporary scholars also see it as an increasingly important domain of inquiry. Some see potential for what they call design thinking to transform the world in many positive ways. Nigel Cross, in the 1982 article “Designerly Ways of Knowing,” sought to distinguish design thinking as a particular domain of knowledge, what he calls “Design with a capital D” (p. 17). Cross argues for design thinking to be explicitly addressed within our educational and social systems, in addition to, and separate from, the sciences and humanities, as a “third culture”. (p.17) The focus on design thinking has seen a resurgence in contemporary academic discourse around technology design with various scholars calling for design thinking as “the next human literacy” (Pacione, 2010, p.7; see also Brown, 2008). Design, these scholars argue, is as foundational as reading and writing, and everyone should be conceived of as designers, purposely trained to engage in design thinking and actively involved in organizing and building systems and relationships to deal with complex
problems and situations in contemporary society (see Pacione, 2010 for a brief review of this perspective on design). In this view, being design literate is something distinct from being an expert or professional designer. Rather, design literacy is about cultivating “basic skills in inquiry, evaluation, ideation, sketching, and prototyping” (Pacione, 2010, p. 9). One does not need special equipment or specialized forms of knowledge to think and act as a designer.

What then counts as design thinking? In other words, what are the characteristics of design as a particular practice and way of orienting to the world? Some relatively consistent answers to these questions appear in various studies of what Cross (1982) calls “designerly” orientations and practices. One theme across this literature is that design in some basic sense is about fitting provisional solutions to situational problems. Tim Brown (2008) speaks of this when he defines design as “a discipline that uses the designer’s sensibility and methods to match people’s needs with what is technologically feasible [and economically viable]” (p. 2). Design thinkers must be attuned to the needs and possibilities of their surroundings.

Various other themes emerge to provide a particular identity to the designerly orientation and practice. Cross (1982) includes the following elements in his definition of design:

- The central concern of Design is ‘the conception and realization of new things.’

- It encompasses the appreciation of ‘material culture’ and the application of ‘the arts of planning, inventing, making and doing.’

---

1 Here I will focus on characteristics of design, rather than characteristics of designers. Cross (1982) and Brown (2008) are among those who provide characteristics of designers in an attempt to understand what constitutes designerly ways of knowing. While I think these are informative for understanding design, I exclude them directly here so as not to privilege the pre-existing characteristics but to instead focus attention on how designerly orientations and practices are attributes of communicative action.
• At its core is the ‘language’ of ‘modelling’; it is possible to develop students’ aptitudes in this ‘language,’ equivalent to aptitudes in the ‘language’ of the sciences (numeracy) and the ‘language’ of humanities (literacy).

• Design has its own distinct ‘things to know, ways of knowing them, and ways of finding out about them.’ (p. 17)

For Cross (1982), realization of things via modeling characterizes a distinct epistemological scope, domain, and methodology of design practice. The orientation to action, or to making something as opposed to reflecting upon some aspect of the world, is a core distinction between a design orientation and other ways of knowing. According to Cross, designers place a primary emphasis on solution rather than problem definition and on the result rather than the rule. Eschewing prolonged analysis of a problem, designers seek a satisfactory rather than optimal solution (Cross). For Cross and others, design tasks have deadlines, or at least time constraints, and “more research” is not often an option. Furthermore, these scholars argue that more research couldn’t provide the answer anyway, since the problems that designers tackle are often ill-defined or “wicked” problems, where all the relevant information to guarantee an optimum solution cannot be known (Cross). The key question for designers, then, is not “what is the exact problem?” but rather, ”what do we build in order to respond to, and improve upon, the problem?”

2 Wicked problems (defined by Rittel & Webber, 1973), are problems that are not open to exhaustive study, since all relevant information can never be available to the problem-solver(s). Design problems are wicked problems, and “every attempt to solve a design problem frames the problem anew” (Ylihisku et al, 2009, p. 1131). Since the goal when dealing with wicked problems is to develop new understandings of the problem domain through good-enough solutions, the process of dealing with wicked problems is ongoing.
Design is thus characterized by a focus on reflection *through* action, rather than action as the result of reflection; the order is reversed in comparison to traditional epistemological frameworks. Design involves a fundamental process of pattern synthesis, rather than of pattern recognition (Cross, 1982). This synthetic approach is constructive and involves actively overlaying codes or patterns onto a problem domain in order to find a solution (Cross). These codes are read and written by designers in “object languages” to “translate abstract requirements into concrete objects” (p. 29). The prototyping that characterizes design practice, in this view, is an ongoing, evolutionary process of constructing artifacts that provide an immediate satisfying solution to a problem while simultaneously generating feedback that helps designers better understand the problem domain. Brown gives the advice that “prototypes should command only as much time, effort, and investment as are needed to generate useful feedback and evolve an idea” (2008, p. 3). Thus, each prototype acts in some way to transform a particular situation of experience. Furthermore, the process of building these prototypes relies upon other technologies, which are used as tools in the design process and play a role in conditioning the situation of experience.

The definition of technology used in this thesis will be very broad to include anything that is patterned or structured and that can be used with some intention as a tool for action. Therefore, the notion of technology extends beyond the purely physical sense of material objects to include things like strategies, stories, texts, skills, theories, rituals, etc., all of which can play some role in coordinating the activities of designers and can also be outcomes of design. Brown (2008) argues that this broader scope of design becomes more visible within contemporary economies due to a shift towards knowledge work. With this shift, increasingly, the products of design are not just physical objects, but rather “they are new sorts of processes, services, IT-
powered interactions, entertainments, and ways of communicating and collaborating” (2008, p.2).

**Design as a communication issue.** The designerly orientations and practices considered above occur within embodied human interaction in particular situations. Basic communication practice, like design, is also distinguished by a reflection through action that can be seen in the prototypical nature of its solutions, which can be understood as “satisficing” ad hoc designs that fit within current circumstances and act to transform them (Cross, 1982, p. 23). The products or outcomes of everyday interaction, in the form of patterns, meanings, relationships, and structures for coordinating interaction, can similarly be described as objects or artifacts that fit our above definition of technology and are capable of transforming experience in complex ways.

To consider the complexity of this relationship between communication and design, I argue that we must see design as it relates to communication in two distinct ways. First, what I described above as design is functionally present within specific communication acts and situations, as actors design new possibilities for interacting with one another and the world. Second, communication more generally is constitutive of design as a practice; communication practically conditions designers and the design process and, in a real sense, shapes the products of design. These two ways that design and communication relate to one another are entangled in practice in ways that make it difficult to talk about them clearly. I will use the phrase “design-in-communication” to denote design happening within communication acts, and the phrase “communication-in-design” to denote the communication happening within design processes.

**Design-in-communication.** Design, according to the definition given above, is characteristic of situated human interaction. Aakhus (2007) takes up this idea in an article that
asks communication scholars to focus on design as a useful way of theorizing and solving problems that occur within the realm of communication. He points out that design is a “natural, describable activity” that can be seen in creative language use and the capacity for humans to coordinate interaction by drawing upon “mutual knowledge and principles of interaction” (Aakhus, p. 113). When we engage in various acts of practical communication, Aakhus argues, we design the possibilities for interactivity on an ongoing basis.

In everyday situations we are often dealing with ill-defined communication problems and engaging in synthetic construction of satisficing solutions. Any number of common sense examples are available that can be understood in this way: managing an interaction when first meeting a person, navigating a culture whose language and ritual are only rudimentarily known, demonstrating some ambiguous quality such as “competence” during a meeting where there is limited explicit feedback to draw upon, etc.. By taking a practical view on how people use language, we can see the process of linguistic interaction as a type of pattern construction, where words and expressions are invoked in order to instantiate abstract requirements for coordinated behavior (e.g. “Look out!”). Because people draw on existing patterned designs (e.g. for how to get someone to pay attention, or for how to sound sarcastic) accessible through our previous social experiences with language use, they can be said to model solutions to local and immediate problems, building upon technologies that were designed or used in previous situations. Receiving feedback about how our designs both solve communication problems and create new ones, we gain a communication design literacy that we develop throughout our lives and apply in new situations.

Design-in-communication is not something that occurs only in special situations or within the formal role of designer. Rather it happens throughout interaction and can be seen in
basic achievements of coordination that we carry out in everyday life. Aakhus (2007) asks us to ground a communication perspective on design in simple collaborative accomplishments of satisficing solutions, such as “people working out how to take turns, what footing to take up with each other, how to open and close interaction, what is relevant, what states of affairs to commit to, and so on.” (p. 113). These accomplishments result despite the fact that we don’t necessarily pay explicit attention to whether or how we’re accomplishing these ends. From the communication perspective, we are designing not only when we purposefully engage in design, but whenever we respond to the situation at hand and solve problems via prototyping of solutions intended to generate feedback about the evolving problem of how to go on within interaction.

**Communication-in-design.** “Design is a way to understand communication”, but it is also “an approach for investigating the social world from the standpoint of communication” (Aakhus, 2007, p. 112). From the communication perspective, the formal design processes that create and stabilize technologies in the world are constituted in and across specific interaction episodes. These episodes of interaction involve achieving the coordination needed before undertaking formal processes of design and production (e.g. Schmidt & Bannon, 1992). This coordination occurs within an environment that shapes the possibilities of action. Yet the structure of this environment is not a given; rather its particular effects are at least partially the result of how it is constructed through communication. Our very experience of this environment itself is shaped, in part, by how we come to look at or make sense of it. The patterns we communicatively construct to overlay upon a situation establish coordination mechanisms that enable formal design. Furthermore, we often do not explicitly define these communicative mechanisms and theories of how the underlying world works, but rather, we imply them through our performance within the situation. The implicit relationships performed within an interaction
episode, according to communication scholars, create certain structures for experience that effectively constrain the possibilities of design (e.g. Shotter, 2003). We can never fully disentangle ourselves from our environment and the coordination mechanisms by which we manage to go on in a world that makes sense. From this perspective, it is impossible to take some objective viewpoint, or to gather objective knowledge, about the requirements, process, and products of design. Then, design is an ongoing practical engagement characterized by designers being embedded in the world. As will be argued below, this embeddedness is reflexive; the designer and the world he or she inhabits are constructed together, at least partially, within the process of interaction. In the resulting view, design is seen less in terms of an intentional process and more in terms of a responsive relational engagement with the world.

**STS and the Relational Boundaries of Design**

Herbert Simon (1969) defines the activity of design as participation by an intentional agent or subject (i.e. designer) in boundary science, where designers work at the interface between the artifact and its external environment. This articulates the relationship between design of an artifact and a fit with the environment of design, what Brown (2008) characterized as a combination of needs, technological possibilities, and economic viability. In considering the communication and interaction taking place at various sites in a design process, however, understanding of the artifact and environment boundaries becomes much more complex, making traditional definitions of design problematic. This is a core insight addressed by scholars in the field of science and technology studies (STS). Scholars in this field are concerned with exploring the knowledge and artifact production practices of scientists and technologists in order to better understand the social nature of knowledge and technology (see Asdal, Brenna & Moser, 2007 for a good overview). A detailed treatment of STS perspectives is beyond the scope of this thesis;
however, I will highlight a few central claims of this literature in order to expand and deepen our understanding of design. The STS literature also can provide insights into how boundaries are constructed communicatively and come to condition the actual design process.

For STS scholars, the situated social practices of organized inquiry condition, in important ways, the presumed objective state of the physical and social world that designers are embedded within (e.g. Haraway, 1988). In this view, the practical, working definitions of objective reality typically used are like prototypes themselves—attempts to overlay patterns upon the world. As a result, our understanding of the objective conditions of the world should involve a theorization of the process of construction whereby the possibilities and constraints are framed.

**Design as negotiation and stabilization.** Within any particular situation, institution, or society, the objectivity that guides design is itself designed, negotiated, and stabilized through interaction. By focusing on how knowledge is produced and embedded in the world, STS scholars illustrate how various relational structures are performed within interaction. In this view, knowledge production necessarily includes setting the terms of conversation, as well as delimiting the roles available to objects and subjects within a particular domain of action. The particular *apparatus* through which we take measure of the world influences the forms taken by the stabilized products of our social and material world and practically limits the possibilities for cooperative action (Barad, 2007).

In focusing on the process and apparatus by which a subject produces social knowledge about the world, STS perspectives challenge traditional understandings of boundaries between what we know and how we come to know, arguing that these boundaries are communicatively constructed and indeterminate in essence. Contemporary scholars in STS thus challenge the
common-sense assumption of technology and science as fields with clear, singular, static, or pre-determined boundaries (Asdal et al., 2007). Rather, they see these boundaries as not only negotiated among humans, but also as embedded in a larger material and social situation (e.g. Latour, 1993). For STS scholars, the surroundings, situation, or context of a particular knowledge production episode condition the possibilities for what forms of knowledge can be produced.

From an STS perspective, it is a highly contingent and particular interaction process that enables the stabilizing of a particular grammar or language for understanding via some apparatus. An apparatus, in this view, is a particular medium capable of translating the world into a particular type of representation available for systematic processing. The apparatus limits the types of data that can be intelligible as knowledge production. On a larger scale, the various apparatuses of social and cultural norms play a role in deciding which phenomena are legitimate interlocutors with a voice, capable of saying something within knowledge production.

STS scholars ask us to expand our view of potentially legitimate interlocutors to involve nonhuman objects, technologies, texts, and environment. A well-known example of this is found in actor-network theory and similar perspectives, sometimes collectively called material-semiotic approaches (Law, 2007). These perspectives argue that nonhuman objects are not \textit{a priori} entities in the world acted upon by knowledge production, but rather can be better seen as “an effect of stable arrays or networks of relations” (Law, 2002, p. 91).\textsuperscript{3} Because they do not exist outside of their relational stability, objects are not simply intermediaries that serve as passive recipients of intentional human design at or of their boundaries. Rather, they should instead be

\textsuperscript{3} Non-human objects include artifacts, texts, discourses, and other forms via our broad definition of “technologies” above, but also include material ”environment” not designed by humans per se.
seen as mediators, playing an active, transformative role in interaction (Latour, 2005b). These same stable arrays are constitutive of the subject, and thus designers are themselves constituted by the nonhuman environment within which they act, at least insofar as this environment can resist human agency. From the point of view of social analysis, humans and nonhumans are symmetrical, as they act together in cooperatively creating social and material realities (Latour, 1993).

In the actor-network perspective, the design process is characterized by many heterogeneous associations and translations that structurally simplify a complex relational field into a particular form called an actor-network (Callon, 1986). This translation is described as a process involving enrollment of people, texts, and objects via mechanisms that stabilize an actor-network by reducing its unpredictability. Stabilization is never complete; it requires a multiplicity of mediators, each playing an active and dynamic role in stabilizing and coordinating the functions of the network.

**Infrastructure and background.** However, the performance that we see is not the whole story, as Susan Leigh Star (1991) points out. Against a theorization that focuses on the executive power involved in stabilizing networks, Star asks us to consider the invisible work involved in stabilizing structure in social and material relations. This invisible work is not captured in notions such as translation and enrollment, which privilege the achievement of actor-networks from the perspective of the actors who are seen as performing these actions (Star, 1991). As Star and other scholars point out (e.g. Haraway, 1988), power and privilege are involved in many hidden ways in the construction of a coherent network of material and representational entities. Instead of focusing on the entities involved in production, we might instead focus directly on the performativity of the surroundings within which production occurs.
For Star (1999) and her collaborators (e.g. Star & Bowker, 2006; Star & Ruhleder, 1996) this means considering the role played by *infrastructure* within sociotechnical practices.

The view put forth in Star’s (1999) work is that the material and social environment is engaged not only in terms of objects and artifacts, but also as infrastructure that provides a supporting backdrop for human action. Infrastructure is experienced in radically different ways across different situations and different actors; what counts as supportive infrastructure for one person, e.g. stairs to move up or down, is prohibitive infrastructure for another person, e.g. one who uses a wheelchair (Star, 1999; Star & Bowker, 2006). Star argues that the mundane infrastructure that “permeates all functions” is not studied widely enough in social theory (1999, p. 379).

Infrastructure, for Star and Ruhleder (1996), is characterized by a quality of embeddedness, or being sunk into environments. As a result of this embeddedness, infrastructure is not necessarily distinguished in terms of its working parts and largely operates transparently, not requiring intentional assembly in order to function. Furthermore, Star and Ruhleder suggest that infrastructure is distinguished by its reach or scope; it is something sunk into a shared environment that has effects beyond a single event or situation of action. Infrastructure shapes and is shaped by conventions of practice, as human beings are socialized within it. Over time, coherent networks that have stabilized can become infrastructure, as various technologies become part of the substrate of an environment that conditions action. Much of the infrastructure that we rely upon in practice remains invisible in our daily lives, and this invisibility is an important dimension of how we relate ecologically to our surroundings, according to these authors.
Star and Ruhleder (1996) point out that all infrastructure is relational; that is, some particular infrastructures are actively drawn upon at particular sites of human activity and practice. There are many heterogeneous layers of infrastructure, none of which is static or outside of context. In this view, infrastructures can be seen to participate in ongoing interactions, both as affordances that enable certain types of action and forms of production and as constraints or obdurate influences that make change and adaptation along a particular dimension more difficult or impossible (e.g. Star, 1999). However, despite a certain inertia, infrastructures remain open to possible transformation or effective displacement as other levels or layers of infrastructure cause changes in the affordances and constraints of the larger environment (Star & Bowker, 2006). Thus, infrastructure can be viewed as a “dense interwoven fabric” that is dynamic, ecological, and even “fragile” (Bucciarelli, 1994, cited in Star, 1999, p. 377).

As a result, Star & Bowker (2006) argue for a more situated rather than universal orientation to the practice of developing and maintaining, as well as reflexively understanding, infrastructure and its relation to human practices. Scholars in STS more generally point to the importance of reflection upon the historical contingency of every layer within practices, objects, and infrastructure, realizing that things might have been different under different circumstances or within different social norms (e.g. Haraway, 1988). Therefore, previous choices condition our surroundings and the possibilities for action. As an implication of this view of science and technology as contingently constructed, scholars in STS call for more participatory engagement in these figurative processes whereby we define the entities of our social world (e.g. Latour, 2005a). Action involves an ecology of mediators, yet we have no grounds to objectively describe this ecology, thus it is important to both critically assess and participate in processes of operationally defining our ecological surroundings (Latour, 1998).
**Communication as embedded design.** In STS perspectives described above, we see knowledge production as an emergence of some structural stabilities via a negotiated coordination among particular words, concepts, and systems, objects in the world, and the practice of relating these together in communication and action. Understanding design in terms of embeddedness within a dynamic and heterogeneous environment leads us to problematize the intentionality of the designer and the process of translating abstract requirements into codes to create an artifact or technology. In fact, these approaches radically decenter the notion of intentionality, reconstructing humans as embodied actors playing only a partial role in design, embedded in fields of relations that are constructed by a multiplicity of mediators. Our interaction is embodied within our surroundings in ways that interfere with stabilization of design practice or communication practice as fully systematic and boundaried languages and ontologies, furthermore. Because of the reflexive relationship between the experience of being in the world and the process of articulating the meaning and structure of this experience, it is a mistake to consider any particular definition of communication or design practice as foundational.

There is reflexivity involved in practicing communication-in-design, whereby choices made within a situation that draw upon social and material infrastructures come to condition that infrastructure and the possibilities for action in the future. Scholars in communication studies use the concept of *intersubjectivity* to address this reflexivity. By focusing on communicational utterances as embedded relational structures, the nature of meaning can be defined as both situational and inherently interdependent (e.g. Shotter, 2003). This leads to a concept of communication as constitutive of the particular structures that enable coordination and intelligibility in a given environment. In looking beyond the process of intentional transmission
of preformed messages to considering the emergence of a particular shape that constitutes infrastructure for action, communication scholars give us an understanding of the surroundings we draw upon and simultaneously create as we coordinate action with one another. A few theoretical depictions of these processes will be discussed in more depth in the next chapter.

As a result of seeing the definitions of our social and material world as constructed in communication, the reflexivity of communication-in-design becomes a central concern of social theory. This resituates design practice as something more complex than simply intentional pattern construction. Designers are always embedded in surroundings, which we perceive as patterned in terms of affordances and constraints, via certain intelligible grammars, and we are always adapting to the context that emerges around us (Shotter, 2006). Design is not a separate act of discerning possibilities and testing one solution, rather, it is a part of a dynamic lived experience within which we are constantly designing, testing, and iterating in relation to largely invisible surroundings. In this view, design is characterized by negotiation and coordination and occurs via relational performances that involve not only human subjects but also various objects and infrastructure. The very notion of a coherent problem situation that design practice can tackle is itself a communicative achievement that results from complex design work of this more basic type.

Communication theories offer a variety of ways to think about the infrastructure of communication practice and of the ecologies it enacts and operates within. However, a detailed treatment of communication theory through such a lens is beyond the scope of this thesis. For our purposes here, it is important to see that these communication practices play a crucial role in the performance of relationships under the expanded problematic outlined above. Concurrently,
the embeddedness of communication practice in surroundings means that communicative acts are not themselves separate from this ongoing process of coconstruction.

**Design in Human-Computer Interaction**

Both aforementioned relationships between design and communication can be studied in the realm of communication technology design. Here we have a realm where theories of coordination become instantiated in technological artifacts that apply particular affordances and constraints to users and, over time, come to act as infrastructure for interaction. Because computation can be seen as a particular type of infrastructure for thought and action, it is a site where the reflexivity of interaction theories can be engaged directly.

Scholars take up these issues directly in the field of human-computer interaction (HCI), and particularly in the related field of computer-supported cooperative work (CSCW). In the past thirty years, work in these fields has addressed the embodied and situated nature of action in the design of technologies that support practice.\(^4\) This work has implications for design-in-communication, as seen in the attempts to design communication technology that supports situated and embodied interaction by focusing on articulation of mechanisms that coordinate interaction (Schmidt & Bannon, 1992). Scholars considering the practices of information workers discuss various issues in supporting practice, calling for participatory and user-centered design of the various infrastructures of information and communication technology (e.g., Greenbaum & Kyng, 1991; Kyng & Mathiassen, 1997).

---

\(^4\) In this context, “support” is a limiting term. There are multiple ways technology relates to the embodied communication that design aims to understand as a starting point: to “convert”, to “augment”, or to “transform” communication can be the results of a particular technology’s intervention (Jackson, 2010).
The need to consider characteristics of embodied interaction applies to supporting not just information work, but to design aspects of this work, and design work more generally. This is seen primarily in end-user development perspectives that seek to provide nonspecialist users with the capabilities for modeling and programming computational systems (Fischer & Giaccardi, 2006). This involves consideration of not only the tools required but also the architectures that support user-designers in action (e.g. Lieberman, Paterno, Klann & Wulf, 2006). Perspectives on participatory design and metadesign environments seek to account for the context and other particularities of these situated actors as they play a role in the development of technology (e.g. Fischer & Giaccardi, 2006; Kyng & Mathiassen, 1997; Winograd, 1995).

**Situated action and technology design.** Two important works in the late 1980s, and the subsequent discussions they raised, are useful to frame the consideration of the HCI field in terms of the perspectives outlined above. Suchman (1987b) and Winograd and Flores (1987) take up the insights of phenomenology of experience and apply these to rethink the design of computational systems, particularly in considering the embodied human beings who are relating with them. It is relevant to this thesis to briefly explore the discussion between these authors, whose works have had a major impact on the contemporary HCI community.

Just as principles of interaction *invisibly* shape the design of communication in order to move forward in a particular situation, the affordances of objects, and social fields that we engage with moment-by-moment, are largely apprehended tacitly or precognitively. This point is made by Winograd and Flores (1987), who draw on the phenomenology of Martin Heidegger and the concept of “throwness” in order to show how what we’re calling design work occurs
within a situation that we are thrown into, with an existing set of tacit affordances and where the
need to act is the basis of our engagement. 5 We cannot separate ourselves from this by
reflection; instead, all reflection and theorizing relies upon this engaged relationship with
experienced surroundings. In action, we do not need to theorize and account for our engagement
with the world; such explanatory structures are added after the fact. As Cross (1982) points out,
you don’t need to know principles of mechanics to know that an axe affords splitting wood.
Instead, we draw from a tacit understanding of how the world works in perceiving affordances of
our environment and surroundings.

Lucy Suchman (1987b) uses the theoretical perspective of ethnomethodology in making a
similar argument about the nature of interaction. For human participants, it is not cognitive plans
that determine our action, and action thus cannot be predicted from knowledge. Instead, she
argues, while a course of action may be retrospectively reconstructed in terms of prior intentions,
“the prescriptive significance of intentions for situated action is inherently vague” (p. 27). She
argues that action is tied not to individual predispositions or conventional rules but “to local
interactions contingent on actor’s particular circumstances” (p. 28). The circumstances of
particular surroundings are central to determining action. We can never fully represent these
circumstances in terms of plans; thus, background or implicit knowledge can be enumerated
infinitely, according to this view. Therefore, Suchman argues, the assembling of intentions and
circumstances as descriptions and accounts of action is done in an ad hoc and selective manner.
The relevance of some particular aspect over another is constructed in performance.

5 While there are numerous points on which contemporary scholarship questions, critiques, and expands
Heidegger’s notion as used here, a treatment of them is beyond the scope of this project. Instead, I’ll consider some
form of embeddedness of communication within preconscious experience of the world as a general design
constraint.
Suchman's (1987b) argument challenges the view that background knowledge is somehow “out there” but remains unarticulated. Rather, we have what Suchman calls a “continually receding horizon of understanding,” whereby the task of accounting is itself a new action, generating new content to be accounted for (p. 46). We can never catch up to our action in our attempts to observe, understand, or predict it. Suchman argues that mutual intelligibility is an achievement not of language referencing to a shared view of the world preexisting the interaction, but rather of an indexical reference of language to situation particulars, which it “presupposes, produces, and describes” (p. 50). Following Heidegger, she suggests that when action is proceeding smoothly, we do not see it “explicitly manifest as goal-oriented activity” (p.53). Only when we need to repair, stabilize, or at least understand something about a process do we treat it as something for which we can conceive procedures and rules.

Since computers are generally thought of as procedural and rule-based machines, this argument has important implications for how we should consider computers in relation to human action. With this, Suchman (1987b) and Winograd and Flores (1987) parallel others in communication studies who challenge a mechanistic view of interaction, a view which was dominant within the field of Artificial Intelligence and much of HCI at the time. The mechanistic view of interaction was grounded in the assumption that the proper application of rational principles (i.e. of “communicative rationality”) would produce machines that could think and communicate like humans. If the affordances and constraints present in collective forms of work cannot be specified a priori, but rather they emerge as a result of communication, how can we

---

6 The notion of indexicality is rich with connection to this consideration of coordination; however, a treatment of it is beyond the scope of this thesis. I will leave this point for future work.
approach the goals of design when we consider the impossibility of design to represent all future interaction possibilities?

Concluding that computers cannot understand the complex structuring effects of speech, Winograd and Flores (1987) proposed encouraging a reflective orientation in users, giving them tools to explicitly communicate the structure of their relational embeddedness. This involves surfacing the relationships implied in communicative acts in order to improve coordination among various actors via the reflective orientation that results. Winograd and Flores argued that we can better account for our throwness by designing systems that allow users to coordinate their interaction through reflecting directly upon the effects that conversations have to the practical needs of a situation. Their aim is that the commitments to action implied in communicative utterances be enabled to act as structuring elements, specifying explicit relational dimensions of communicative acts so that action can be better coordinated. Using speech act theory (e.g. Searle, 1969), they present the COORDINATOR, a system intended to improve coordination by making interactions transparent in terms of their actionable dimension. This involves reflectively specifying an often tacit pragmatic dimension of communicative acts and relationships. This pragmatic dimension can only be understood from within an interaction, they argue, but can be articulated and thus used to structure a computer representation of interaction.

While acknowledging the important conceptual space opened within technology design by Winograd and Flores, their contemporary Lucy Suchman (1987a) nonetheless remains highly critical of their attempt to create abstract and generalized models of the pragmatics of human communication. In an essay reviewing their book, and in her own book, she argues that actions are irreducibly situated in contexts and that plans and theories (including all ordered understandings of practice, such as formal communication or phenomenological theory) are not
universal metaphysical, cognitive, or social structures that determine our behavior. Instead, drawing upon the insights of Garfinkel’s (1967) ethnomethodology, she sees theories and frameworks as “interactional deployments” to account for action after the fact and make it meaningful (1987b). On this view, situated interaction is not reducible to any formalized model, even a supposedly context-sensitive one such as that put forth by Winograd and Flores (1987).

Suchman (1994a) argues that any attempt to impose a “language-action ordering,” or a singular interpretive framework of any sort, upon interaction raises the question of “how our relations to each other are ordered and by whom” (p. 188). For Suchman, the context of an interaction is the edifice on which it rests, and the structure of this context is built from the ground up in every conversation. Where Winograd and Flores (1987) go wrong is not in attempting to encourage a reflective orientation and allow action in some way to speak structurally, nor is it in their particular choice of framework. Rather, they go wrong in applying their model unreflexively in organizational contexts where a dominant discourse uses language in deterministic ways. For Suchman (1994a), designing a singular, institutionally-deployed and top-down, nondiscretionary determinate system for encoding language and conversation, regardless of how intelligently and wisely designed it may be, is a problematic approach because it tends to create a singular ontological framework in practice even if it doesn’t presuppose or require one in theory. This argument aligns with perspectives from STS that thematize the role of power in determining whose values are instantiated in design and knowledge-production practice (e.g. Haraway, 1988).

These authors agree that there is a deeply reflexive relation between artifacts and their circumstances of production. However, for Suchman (1994a; 1994b), seeking foundational understandings of communication leads to seeing a tool as neutral and thus, ignoring important
political aspects of its context. A local experience includes many invisible (implicit, unknown, obfuscated) elements: emotion, conscious and unconscious intention, or belief, for example. Suchman (1994a) argues that if we are to seriously honor the place of situated action as uncategorizable via representations in sense-making systems, we must acknowledge incoherence and plurality, as well as respect the primacy of experience, within our designs.

**Second and third wave tensions.** These conversations inform the second and third waves of research in HCI, roughly corresponding to the 1990s and 2000s respectively. Calls to attend to the importance of situated action in design led to what has been called a postcognitive second wave of HCI and CSCW that focused on dynamic situated coordination. Schmidt and Bannon (1992), in an influential second wave paper, talk about the problem of supporting articulation work, the “meshing” of tasks, efforts, and actors, that are characteristic of cooperation in their view (p. 18-19). They describe the key issue in CSCW as “how to support the ongoing dynamic articulation of distributed activities and the cooperative management of the mechanisms of interaction themselves” (p. 20).

While the second wave challenged cognitivistic and mechanistic biases, its focus on functional effectiveness of general systems led to a drive to theorize foundational understandings of practice at the expense of the more particular aspects of the direct experience of interactants. Following Suchman’s critique above, this insight is a starting point for a third wave in HCI that both extends and pushes back upon the work of the second wave (Bødker, 2006). According to Bødker, the third wave extends the second in terms of the breadth of contexts under consideration, moves away from the goal of foundational understandings of situated practice in general, and lessens the emphasis on a functionalist orientation in technology design. The third
wave explicitly demands systems that are open to interpretive flexibility and supportive of user meaning-making (Sengers & Gaver, 2006).

Whereas second wave systems are generally assessed in terms of effectiveness, third wave systems seek to open up new experiential possibilities, promote reflection, and generate emotional experience in unexpected ways (Bødker, 2006). Bødker argues that prototyping experiments must be performed not simply to test solutions but to encourage reflexivity and promote asking the right questions. Furthermore, she calls for HCI scholars to expand the focus on boundary work, and to consider the need to accommodate multiplicity by combining multiple levels of tailoring in order to integrate components and mediators that originate “outside of the context of the immediate configuration of mediators” (p. 6). Bødker asks designers to focus on supporting the cooperative process that are involved in such reconfiguration of mediators.

Here the focus shifts from specifying communication technologies that support complex interaction to the broader aim of building systems that support ongoing, flexible, and evolving design practices. Rather than figuring out and implementing the best infrastructure to support situated action in general, the focus here is on supporting and adapting to the process and context of a particular interaction situation. For example, Sengers and Gaver (2006) provide a rationale for purposefully designing for flexibility and describe strategies for designing with the goal of multiple interpretations, stating that “if we take supporting multiple interpretations as a central goal, design shifts from deciding on and communicating an interpretation to supporting and intervening in the processes of designer, system, user, and community meaning-making” (p. 102). They argue that explicitly recognizing the legitimacy of multiple interpretations has fundamental implications for both the process of design and the process of evaluation. As a result of privileging a situation of interaction over a static understanding of infrastructure in
considering design of computer systems, the problematic includes requirements for systems that act over time to emerge, adapt to situations, and evolve within changing ecologies. The major point is that the problem is one of ongoing and participatory design.

**Continuous redesign.** The need for ongoing redesign must be considered as an infrastructural problem at the level of software structure itself. The paradigm of conscientious software is a promising approach that expands the theorization of how systems might be designed for open-ended evolution (Gabriel & Goldman, 2006). Gabriel and Goldman argue that it impossible to capture interactions in a static specification that enables thorough testing due to the sheer complexity of interactions, unanticipated interactions (both direct and indirect), and other “implementation combinatorics” (p. 2). This practical orientation resonates with many of the perspectives described above and broadens the focus to consider how computer systems might be designed to coevolve with situational human needs. They thus argue for a change in the location of design:

Design is less a result of pre hoc planning than of slowly dawning insights—insights derived from seeing how the thing turns out and is used. Continuous (re)design will move out into the field because the field is where the observations are immediate and changes can be tested rapidly *in situ.* (Gabriel & Goldman, 1996, p.2)

Toward accomplishing more robust ongoing redesign, the authors highlight the analytical and practical value of distinguishing between two levels of systems. First, allopoietic systems are described as those that operate deterministically within the problem domain as encoded in a given system, running algorithms and performing functional transformation to provide the content of the system and enforce its rules (Gabriel & Goldman, 2006). Second, the autopoietic
components are those that function to monitor and modify the allopoietic parts to improve or update their functionality, providing interfaces to the context of the system’s operation, enabling the modification and repair of the allopoietic mechanisms (Gabriel & Goldman). These authors argue that this self-correcting dimension should be incorporated into software structure itself in order to better manage the evolution of software but should generally be separated from the internal capacities of a particular software module, enabling reuse and adaptability of components.

Gabriel and Goldman's (2006) approach places redesign within the essence of the software program, rather than treating it as disconnected from allopoietic systems. They argue that while many approaches to software design have attempted to engage with the end-users during the design and development process, these approaches tend to focus on the designer as an agent whose goal is “getting things right” during the initial design phases (p.2; see also Sumner & Stolze, 1997). This view, they point out, constrains evolution and change of the technologies to the “factories” where they are initially designed and produced. In contrast, their approach coincides with others calling for ongoing design in moving the focus away from initial design and toward ongoing negotiation and adaptation of software within a particular environment.

As the requirements of use surrounding software change over time, these authors argue that software must become better at establishing interoperability and forming new relationships with other users, systems, and infrastructures. Gabriel and Goldman (2006) describe an ecology of components where rules governing interoperability between components are not predefined but are negotiated in situ as well. While systems must be robust and self-contained, capable of maintaining integrity across these changes in environment, their interfaces must also be more accommodating to various possibilities of fitting with the environment.
Gabriel and Goldman (2006) describe what this sort of relational computing might look like:

Software will be constructed from components that want to form a community. Systems will be components—either within the same address space or not—that interact as needed, even supposedly/originally monolithic systems. This will enable us to treat every system as if it were a distributed system, which might mean that systems can be more robust to single-point failures. (p. 3)

The above demonstrates a few of the ways in which CSCW scholarship has approached designing computational systems to support complex, situated interaction. In this literature we see both an increasingly complex understanding of situated action and a focus on designing for open-ended and context-driven redesign. However, this literature has only begun to address the demands of participatory, ongoing design of software in sociocultural terms at the level of general infrastructure for communicative practices. Whether existing collaboration systems can support the reflexive entanglement of communication and design in the ways outlined above is an open question that this thesis seeks to address.

**Reflexively Theorizing Within a Metamodel**

Let’s summarize the expanded view of design that we’ve begun to unpack in this chapter. Traditional perspectives on design consider design work as an engagement with the practical world, however they take the practice of design itself as relatively transparent in terms of its intentionality being situated in the designer acting upon objects that are out in the world. The more complex view here considers the designer, the problem domain, and the surroundings as all constituted in the process of design, rather than pre-existing it in a straightforward, knowable,
way. We see design as a communicational phenomenon, understanding knowledge production as accounting that influences, rather than simply describes, the world. As designers, we and our objects of design are relationally embedded in surroundings that are continuously evolving through ongoing interaction, and our processes of thinking and communicating are inescapably entangled with the surroundings.

Seen as patterns overlaid upon the world that we continuously adjust via feedback to fit with an evolving understanding of a particular domain, all explicit theorizing can be characterized as design in this sense. How people theorize and develop shared understandings of the world, how we make sense of our embeddedness and the possibilities for engagement in the world as designers, shapes communication and design practice. Our established languages for describing the world prefigure the shape of the domain we seek to understand. If we take the communication as the phenomenon we want to understand via a designerly inquiry, the reflexive relationship between our existing theories of communication and the outcomes of design becomes a practical one.

As Aakhus suggests, communication theory takes on a practical dimension in design research, as we ask how people will behave if they understand a particular theory (2007). He points out that, even at the most mundane level, each communication design contains implicit and/or explicit hypotheses about how communication works and ought to work. Designers who build actual technological artifacts have an important influence in shaping, conditioning, and disciplining communication in society (Aakhus, 2007). If designers believe this or that particular theory about communication, how will these theories become instantiated in designs?
The debate outlined above within the academic literature between Suchman and Winograd highlights tensions around the use of foundational theories of embodied interaction within design of communication technology. On the one hand, every design encodes some sort of foundations. These foundations, even when not explicitly defined with regards to communication theory, become instantiated into technological artifacts that act as conditioning structures on experience. In the realm of HCI, interaction theories define intelligibility for a machine that must respond to human and environmental inputs in selective, procedural, path-dependent ways. Given useful foundational theories that fit with a particular environment, alongside clever, thoughtful design and effective communication among designers, the products of design can be robust and flexible systems to support communication and coordination.

However, a highly contextual and constitutive understanding of interaction argues that we must always treat foundational theories not as underlying realities but as practical deployments within a larger ecology of experience. Theories and technologies that enact them are only as deterministic as the power vested in them through social performances of various sorts. In this sense, an indeterminate context of practical experience is the foundation of action and interaction, and the particular structure of understanding emerges from the ground up, in ways that cannot be prespecified in categories under theoretical mappings of action.

**Communication as constitutive metamodel.** Robert Craig (1999) argues that practically speaking, communication theorizing is performed by everyone. This is seen in the practice of metadiscourse, or communication about communication. In order to clarify and interpret communicative acts, we engage in practical metadiscourse about the process of communication, actively constructing a theoretical understanding of it in order to solve immediate problems of coordination. This is not just the realm of scholars; rather, it is already
present in the “lifeworld”, where communication is a term within shared social histories that is rich with meaning (p. 120). What ties the diversity of practical communication theory together, for Craig is the use of a “communicational” perspective that suggests processes of communication to explain or account for the existence of some aspects of reality (p. 124).

Defining communication theory practically, literally by its practice writ broadly, Craig sees a “coherent field of metadiscursive practice”, whereby people talk about their communication, with reflexive “implications for the practice of communication” (Craig, 1999, p.120). Craig states:

Communication theories are reflexive: Formal theories, that is, often draw from ordinary, culturally based ways of thinking about communication but these theories, once formulated, can also influence, either to reinforce or to change, everyday thinking and practice. The relationship between theory and culture is thus reflexive, or mutually constitutive. (p. 125)

Metadiscourse has a reflexive effect on communication practice not only in mundane contexts of everyday life, as Craig points out, but also among scholarly theorizing about communication. Among scholars developing and using communication theory, the practical role of metadiscourse is also present. As Shotter (2008) points out, any theory must be accompanied by a dialogic account of how the terms it sets out should be understood, an instruction manual or practical metatheory. Formal communication theory is “theoretical metadiscourse” for Craig, and is conditioned by the “practical metadiscourse of everyday life” (p. 124). No communication theory can fully describe the foundations of interaction, since theory itself requires a dialogic negotiation of shared understanding about how it is to be treated in relation to
a particular situation. This process of negotiation, while it could be retrospectively understood in terms of the theory, occurs within a practical context of action that conditions how the theory is deployed.

Craig is interested in applying this insight as a means to bridge the various models of communication so that they are better able to talk to one another, within a framework that does not undermine the conceptual irreducibility of each particular theorization of communication. Craig suggests a “dialogical-dialectical” coherence across different particular theorizations, defining this as, “a common awareness of certain complementarities and tensions among different types of communication theory, so it is commonly understood that these different types of theory cannot legitimately develop in total isolation from each other but must engage each other in argument” (Craig, 1999, p. 124). A particular communication theory must be able to explain and account for itself, at least partially in relation to other possible theories and perspectives.

Craig addresses this directly by proposing a metamodel that treats communication as constitutive, arguing that such a metamodel “opens up a conceptual space in which diverse first-order models interact” (1999, p. 126). In contemporary communication literature, the constitutive model of communication as a phenomenon is differentiated from the transmission model, whereby we treat communication as a linear process where singularly bounded entities send and receive similarly bounded messages. As should be clear by this point, perspectives on embeddedness parallel and draw upon this constitutive view of interaction. However, any particular constitutive model is itself a first-order model, and Craig takes a pragmatic view that transmission models can coexist with various constitutive perspectives within the larger field of communicational explanations if we provide this field with conceptual infrastructure for
conversation. To do this, Craig suggests that we categorically separate a general constitutive metamodel from first-order models, while nonetheless acknowledging that any particular metamodel is paradoxically also a first-order model and therefore can never be completely neutral. From a practical perspective, however, Craig argues that it is valuable to develop a metamodel so that conversations about the nature of communication can be coherently carried out. He argues that without a metamodel providing some coherence to the domain of inquiry, the phenomenon of “communication itself” can be confused in conversation with the term as defined in some limited tradition of communication study (p. 127).

**Extending a metamodel to technology design.** Asking how communication is constitutive of social reality gives us some ground to engage with the complementarities and tensions of not only academic theorizing, but its reflexive, practical implications as well. Attempting to answer this question from a designerly perspective would involve building models of constitutive communication and overlaying them upon the world, in order to see how they engender different practices of communication. Craig’s work develops a detailed metamodel that applies to the domain of formal communication theory, in an attempt to bring disciplinary coherence to a field widely cited as lacking it. For the academic discipline of communication, Craig undertakes to design a metamodel, distinguishing and proposing relationships between various traditions of communication theory in order to create coherence. In this thesis I seek to actively appropriate the spirit and basic ideas of Craig’s work and apply them to technology design more generally.

To begin, I note parallels between his practical definition of communication and the definition of design developed in this chapter. Already happening everywhere in everyday interaction, design is like communication in that it is practiced by everyone, and that it is
practically theorized when we reflect upon how we do it. Academic research and educational practice that defines and describes design, and attempts to promote expertise in it, and aims to design tools and environments for designers, parallels communication theory as described by Craig. The language of design, broadly speaking, is similarly “metadiscourse, a discourse about discourse in the context of a practical discipline” (Craig, p. 128). Theorizing design, like theorizing communication, has a reflexive relationship to its practice, as it encodes structure into the world that influences future thinking and practice. In this view, we might define the ongoing process of reflexive design as metadesign about design in the context of a practical discipline. In both design and communication, this reflexivity is particularly seen within the design of technology to support these very processes.

Where Craig proposes that a constitutive metamodel can create basic dialogical-dialectical coherence within communication theory as a field, in this thesis I seek to extend his argument to apply to the design of communication technology that supports ongoing design. Could we build a metamodel that aims to define a common ground for intelligibility across first-order communication and design processes, despite the unique and irreducible structures of each different situation? Can we build such a thing not only as conceptual technology, but as a computational system capable of dealing with the reflexivity of its own relationship to first-order models of interaction? My basic argument is that a practical understanding of communication as constitutive can serve as a guideline for design and ongoing redesign of a computational architecture that prototypes this metamodel. By modeling, albeit selectively, the reflexive relational embeddedness that is a characteristic of all situated action, a technology might be developed that creates a base of affordances for intelligibility and interoperability not only
among objects of design, but also among first-order models of communication emerging in local processes of design.

Such a metasystem would be designed to afford the establishment of dialogical-dialectical coherence through coordination and negotiation of some common ground for intelligibility. Yet it could not be measured in terms of objective effectiveness, since any measures of intelligibility would necessarily emerge within the particular situation in which it is being used. Since the inexorable process of embodied action continues, its nature is dynamic and indeterminate, and its design is ongoing, whether or not any particular interaction can be coordinated successfully. Boundaries that it draws would not be static and permanent, but dynamic and shifting, and would presumably require communicative action in order to be stabilized. The implication of this for design is that we might re-orient the designer’s purpose away from defining coherence a priori, and towards somehow enabling a metasystem to develop situational achievement of communicational coherence in collaboration with particular users in a particular situation.

So then I ask: how to design such a metamodel? Because of the reflexive relationship of practical theorizing to the structure of our world, and because theorizing occurs within the stream of embedded, responsive experience, the metamodel of communication as constitutive cannot fully be defined a priori. The various questions raised in this literature review point to the need for stronger theories of communication as constitutive to guide the evolutionary participatory design of a metamodel system. The process of applying theory is reflexive; there is an ongoing coproductive relationship between the particular theories chosen and the relating of these to particular situations through interaction. This is true whether we are discussing theories that are tacit or explicit, and whether they are encoded in computers or in the knowledge of participants.
To meet the criteria of modeling reflexivity, a metamodel must draw upon some foundational theories of interaction to construct a system for modeling possible coordination and coherence, yet it must be simultaneously pragmatic and antifoundationalist in its orientation to adapting its own structure and rules to the uniqueness of any given situation. In the following chapters I will expand and develop the proposed goals of design though an actual design process, and will propose and subsequently analyze the feasibility of a design for a metamodel-driven computer-based interaction environment.

**Conclusion**

In this chapter, we saw that design practice has a complex relationship with the phenomenon of communication, and that this relationship brings out the reflexive connection between experiencing the world and defining the structure of that experience through interaction. I proposed that a view of communication as constitutive can expand the definition of design in productive ways, particularly by extending the concept to include ongoing negotiation of relationships that condition design practice and its outcomes. My intent was to demonstrate that taking seriously a constitutive perspective in communication and design practice calls for grounding our understandings of design within practically-oriented interaction and taking a broader view of communicative action that includes nonhuman objects, surroundings and infrastructure as mediators. Using this ground to develop a metamodel technology capable of providing an infrastructure for negotiating situational intelligibility of relationships, I argue, may be valuable to expand the possibilities of participation in the design of our social and material worlds.

In the next chapter I expand upon the insights of a constitutive perspective on communication in an attempt to outline the boundaries of a prototype system. I consider some of
the attributes that such a prototype should possess, and propose particular ways of modeling these characteristics within a technological artifact. In the third chapter, I assess whether the proposed system is feasible within existing multiuser synchronous communication environments, with a focus on current limitations and issues alongside promising strategies and approaches. I conclude the thesis by examining this design and analysis process in terms of larger and broader questions of whether a metasystem might be feasible and viable as an infrastructural solution to the problem of reflexivity in communication and design.
Chapter Two

Introduction

In the previous chapter, I outlined a particular design problematic and proposed that the notion of a communication metamodel might be a valuable approach to design within this problem domain. In this chapter, I address the design of a computational metamodel directly. Communication as constitutive of relationships and structure provides the core principle around which a prototype was developed. First I will discuss some theoretical perspectives that posit communication as constitutive, giving further shape to this very general idea and proposing some elements of a first-order model of interaction. Along with the resources of the first chapter, I interpret these theories as a designer; the posited constitutive processes are then constructed onto an imagined computational system. Major features of the proposed system architecture are discussed in terms of how they would operate in real-time synchronous use contexts. I conclude by summarizing an informal set of requirements for subsystem functionality and asking the research question: “could a system based in this metamodel be feasibly developed with current infrastructures?”

Craig (1999) points out that a metamodel necessarily involves some first-order model of communication that undergoes a “reflexive turn” (p. 124). Within a metamodel, this first-order model works to allow other models to productively define themselves and relate together via the language provided in the first-order model. Instantiating a metamodel means invoking some particular theory of communication and processing other models and situations through this theory. In what follows, I outline theoretical resources of one particular constitutive theory that
will serve this function in my particular metamodel design. I begin here with a theoretical discussion of some constructs that provide answers to the question of what it means for communication to be constitutive.

**Communication as Constitutive**

Theoretical considerations taken up in this section of the chapter will fall broadly into the following areas: (a) communication as a process of stabilizing reality and constructing boundaries, and (b) communication as emergent through, and constitutive of, responsiveness between participants. Related to both of these considerations are two distinct, yet interrelated, aspects of communication-as-constitutive: (a) the implicit and presuppositional forces of communication acts (and the relation of these to explicit forces) involved in structuring interaction, and (b) the role of time in constitutive communication and the temporal dimension of communication acts. Finally, reflexivity will be discussed briefly as what distinguishes a metamodel design from the first-order model it begins with.

**Communication as boundary construction.** As outlined in the previous chapter, presumptions about communication and the entities of the world implied within particular communicative formulations condition our experience of the real. In communication scholarship, the constitutive view of communication is often situated as a critique of, and alternative to, the transmission view (Craig, 1999). While transmission views treat interaction as a series of discrete exchanges and transfers between fully separate and predefined entities across channels of space, the constitutive view argues that when we interact, we are engaged in forming and re-forming the very boundaries we presuppose. We are already entirely responsive to our surroundings and the situation, and, as a result, we actively separate out entities, and then only partially, which come to appear stable* as a result* of communicative achievements. If we accept
the radical notion that communication is constitutive of the very entities we traditionally describe as actors, we could instead look at these entities as themselves no more than “stable arrays” of relations (Law, 2002). As researchers and designers, we face the problem of understanding the emergence from within experience of the very stabilizations we take to be the foundations of interaction.

Drawing on earlier thinking about emergence of boundaries, contemporary organization scholars are among those who provide a critique of the cultural bias to think in terms of already-formed entities when theorizing the world (e.g. Chia, 1999; Tsoukas & Chia, 2002). In the ways that our biases about already-formed entities prefigure the problem domain of organization theory, these authors see a problem whereby focus on stability tends to presume such stability as existing a priori to the situation. This view obscures what they describe as an inexorable process of change that underlies any given situation, upon which boundary construction and stabilization serve as a strategy to make the world actionable in organized ways. Instead, they argue, we must learn to see differently, in terms of intuition, or in “intellectual sympathy” with our dynamic surroundings (Nayak, 2008, p. 178). In other words, in order to understand processes of sense-making and coming into being, we need theories about the achievement of stability.

**Registration as stabilization.** Achieving stability is not merely an epistemological concern, but has an ontological dimension as well: how do we achieve the stability necessary to understand action in terms of subjects and objects? One process-oriented theorization that addresses the ontological dimension is the conceptualization put forth by Brian Cantwell Smith in the book “On the Origin of Objects” (1996). In this book, Smith provides a detailed theory of

---

7 Philosophers Henri Bergson, Alfred N. Whitehead, and William James are widely cited as the intellectual foundations for contemporary process theories (e.g. Chia, 1999; Nayak, 2008; Shotter, 2006).
the process by which boundaries emerge from the relational dynamics of participation in reality. Smith uses the term *registration* to denote the process of coordination. Relative to a dynamic background of change, registration is an active process performed by some internally-organized system that possesses *intentionality*. To register the world, for Smith, is to “*parse, make sense of as, find there to be, structure, take as being a certain way—even carve the world into...*” (p. 191). For Smith, this notion is broader than, and encompasses, processes that we describe as representation. Unlike representation, Smith argues, registration is not detached completely from the dynamic underlying world, but rather, “reaches through the world,” signifying “successful establishment of an intentional relationship” (p. 194).

Boundary construction, by its nature, requires a particular sort of separation over time that follows after a particular encounter in a moment of complete coupling or connection that Smith describes as “ineffable” (1996, p. 292). Smith argues that registration is a response that occurs in a subsequent disconnection from some particular *region* of the world and serves as a way to preserve the immanence of an experience by establishing mutual coordination of a subject with an object of perception.

For Smith (1996), registration enables coordination with some otherness beyond *effective reach* of embodied sensory connection. In describing registration this way, Smith relies on a distinction between effective reach and intentional or *semantic reach*. The effective reach of an embodied being is defined as the materially structured perceptual boundaries of this proto-entity; traditionally, this is called the senses or sensory organs and perceptual systems, which are themselves constituted and extended through various technological apparatuses. Semantic or intentional reach is defined as a broader form of coordination between an intentional system and its surroundings, as it involves the ability of the system to perform abstractions that relate a
particular event of connection with memories and anticipations of past and future events respectively. These abstractions are what enable an intentional subject, or “s-region,” to re-identify something as the same form as that which preceded it at an earlier point in time. The power of registration, Smith argues, is that it creates a language enabling semantic reach across distances that extend beyond the “right here, right now” of embodied experience. In this view, we construct entities from within a changing world by constructing a boundary around an abstracted region of otherness, or “o-region,” in order to enable future coordination with that effective otherness.

It is in this sense that registration reaches through the world to establish a relationship. Because of the disconnection and abstraction involved in doing so, registration is always only an approximate coordination, characterized by a fundamental uncertainty about the status of its distal object beyond effective reach. After effective reach is broken, the mechanism of intentional reach becomes internally located within an s-region. Registration can be seen as retraction of the mechanism for coordination into the s-region, giving registration a “directedness” that makes it asymmetrical (Smith, 1996, p. 223). In this view, the subject or s-region is primarily, but not solely, responsible for constructing and maintaining the boundaries that give order to a fundamentally dynamic world.

However, this does not mean that everything is subjective. Smith (1996) argues that the gap separating subject and object remains symmetrical and thus registration does not reduce or internalize the o-region to an object in thought. Stabilization of the relation is not one-directional; rather the regions rely upon each other. This symmetrical formation of stable arrays echoes the argument of science and technology studies (STS) theorists, such as Bruno Latour (1993), discussed in the previous chapter.
More specifically, Smith (1996) argues that the reference-preserving action of subjects is, in itself, not enough for stabilization. In order to maintain coordination via registration, an s-region (while it need not be itself registered) must compensate for its own movement through time and space as well as the movement of the o-region. Smith argues that the stability of an s-region reciprocally relies upon the internal structure of o-regions situated in relation to the s-region. An o-region with a particularly stable pattern of internal dynamics, such as a clock, reciprocally helps provide a stable ground for maintaining coordination across changes within an s-region (Smith, p. 258). Put more simply, an object that is structured to recur in relatively predictable or immutable ways in a situation creates a resistance that conditions the s-region and helps to stabilize it.

Thus, o-regions with a characteristic of sameness-across-difference become actors in stabilizing relations, forming a center of gravity that insulates them from changes and enables effective influence on surroundings. This centripetal process is not reliant on the s-region for its agency, although it may require upkeep. Smith states that, “synthesis, maintenance, and repair” of objects also play a role in stabilizing relations (1996, p.264). This echoes Star’s (1999) perspective on infrastructure. It also serves as a reminder that we are not simply designing stabilization mechanisms nor simply relying on preexisting ones, but rather that we are constantly developing, maintaining, and repairing the infrastructure we already invisibly rely upon when developing new infrastructure.

If mechanisms for coupling at a distance are to succeed, they must be fit to their surroundings. As objects successfully become integrated into the environment as infrastructure for the achievement of stabilization, they can also be seen as apparatuses that extend the possibilities of coordination. Apparatuses are infrastructure invisibly coupled to the embodied
subject until a breakdown makes it visible and calls for some repair or until subjects reflexively engage in its development. All technology functions this way according to Smith, including “instruments, skills, theories, inference, and civilization” (1996, p. 224).

We then can see how interaction is characterized by a complex, adaptive struggle to construct active boundaries. As Smith (1996) points out, this perspective posits that it is across the boundary separating subject and object that the stability consists. Thus, in this view, it is communication, in the broader sense of interaction, that provides stability in constructing relations. The insight is that the performance of a relationship creates, or at least reinforces, the stability of the entities at either end of the relationship.

**Communication as emergent responsivity.** Within Smith’s (1996) theorization of coordination is the notion that registration creates commitments. For Smith, effective and semantic reach combine in a commitment to (and participation in) “multiple coincident boundaries” (p. 218). As outlined above, participation in the world involves the whole of an embodied subject, not just their cognitive processes, and Smith argues we are thereby invested in our registrations with our entirety of being and not just our thoughts or words. Acts of registration imply some responsibility to maintain coordination, and the achieved stability that emerges across these performed boundaries partially results from attention to this responsibility.

Perhaps it’s fairly clear how objects can emerge from registration and ongoing coordination, but how do subjects emerge? The above considerations all point to a significant insight: communication is more fundamentally about response than it is about intentional transmission. Intentionality and subjectivity are not outside of the flow, but rather, are constructed in responsiveness to surroundings. Various philosophers of dialogic encounter
explore the nature of connection and relationality within specifically human experience, from a phenomenological perspective. Their insights about the dialogic nature of being enrich our way of understanding communication as constitutive, by positing basic relationality as fundamental to the intelligibility of being.

Philosopher Emmanuel Levinas (1969) sees the singular irreducible element of Being as not located in the unique ownership of subjective and reflexive experience of the self, but instead within an encounter with the “Other” (Hand, 1989, p. 4). For Levinas, the relationship with the Other is enacted in an encounter whereby we are taken outside of our subjectivity, and are inexorably taken across the boundary that separates us from this Other. This relation structures the very experience of being in the world and is irreducible to anything that can be subsumed in the concept of individual consciousness. Martin Buber’s philosophy of dialogue (1970) contains a similar phenomenology, arguing that engagement of the aware self with reality, including any acts of theorizing that constitute the patterned experience of this world, are grounded in moments where the embodied consciousness experiences a fully immersive “standing in relation” to an Other, or a “Thou” in Buber’s terms (p. 55). This is contrasted with experiencing the other as simply an object (an “It” for Buber), a representation generated as an element within our projected self-conscious relation to a contextual reality.

Buber’s phenomenology finds significant parallels in Smith’s notion of participation. Both connection and distanciation are phenomena within the larger process of experience for Buber. Moments of encounter with a Thou are episodic, returning us back to the It orientation of

8 Levinas and Buber have somewhat different views on important matters related to the phenomenon of encounter with Otherness, but a consideration of these is regrettably beyond the scope of this thesis (e.g. Atterton, Calarco, & Friedman, 2004)
interior consciousness. Yet, these encounters similarly remain immanent in consciousness as a potentially resurfacing totalizing connection and thus persist in their effect on the “I”.

It is in transcendence of self that the reality of being in the world is constructed. Levinas and Buber interpret this humanistically, however many contemporary scholars propose post-humanist alternatives (e.g. Clark, 2003; Haraway, 1991, Hayles, 1999; Latour, 1993). In these views, multiple forms of nonhuman otherness can also be considered as participants in transformative encounter—this assumption is implicit and often explicit in the works considered in the preceding chapter. We are in a dialogical engagement with not only single human Others, but with our entire surroundings. This view coincides with a broader, ecological perspective on design.

Another humanistic tradition in dialogue studies, Bakhtinian dialogism, provides valuable insights into the ecological complexity of embeddedness. Bakhtin’s philosophy, addressed in contemporary communication scholarship (e.g. Barge & Little, 2002; Shotter, 2006; 2008) is one attempt to consider a heterogeneous and agential multiplicity of otherness in terms of how this multiplicity comes to structure particular communicative events or utterances. For Bakhtin (1982) the experience of being in the world can be usefully seen as a field or fields of interacting (“polyphonic”) voices and discourses collectively constituting heteroglossia. Each possible voice that is immanent and can be brought to bear on a particular situation is a potential addressee to which communication may be at least partially oriented (in response and in anticipation). The singularity of a particular Self or Other is an illusion, and/or an achievement: subjectivity is composed of many diverse, often conflicting, voices and discourses combining together to structure the experience of self and situation.
Dialogic thinking in terms of a multiplicity of otherness brings the core insights of Levinas and Buber about the dynamics of inter-subjectivity into encounter potential addressees beyond the physically present, embodied human other. These perspectives also extend beyond describing singular processes of registration to consider entire fields of relationships constituted in ongoing interaction. Both registration and dialogism draw on the common concept of a responsivity (or responsibility) to otherness which is not only transformative in its presence, but powerful in its immanent being as well. For each of these thinkers, the interiority of subjectivity is realized through connection with the exteriority of otherness.

**Emergence of structure via implicit presuppositions.** The above theoretical ideas perhaps beg the question: How do our specific communication acts constitute relational structure? If the communicative self is constructed in an encounter with otherness, then the interaction between and among some sort of forces is primary to forming the boundaries of the self. However, since we do not want to presuppose these forces as entities, how might we understand communication as taking place outside of people or things? We seem to have moved far away from actual human language use and communicative expression. How to account for choice and the particularities of human expression? How, for example, do we come to be a semistable subject, actively relating with certain types of otherness while ignoring or effacing different relational possibilities? The theories outlined in this chapter and the last ask us to perhaps understand responsive engagement with a complex field of addressees as something larger than (and constitutive of) the human individual. But then how to understand the apparent communication acts of human individuals? Niklas Luhmann takes an interesting approach, arguing that the phenomenon of communication itself is a relational system that is autonomous from and transcendent to human intentionality. For Luhmann, as for dialogic theorists in the
Bakhtinian tradition, the ambiguities of experience implicitly call us to respond by structuring possibilities for “going on,” drawing us into relations with the world around us (e.g. Shotter, 2003).

Luhmann does not describe responsiveness in terms of particular otherness, but instead sees communication as a self-reproducing generative phenomenon with no “immanent entelechy,” (no inherent goals), that is outside of and autonomous from human construction (1992, p. 304). In his view, the goals or intentionality that underpin a communicatively-constituted situation do not exist at the level of the relational system itself, or, if they do, they are not discoverable and knowable. Coordination mechanisms themselves do not have inherent goals; rather, actual coordination and a sense of intentions emerge from within actual relational performances. According to Luhmann, it is because of a situation's inherent unknowability that ambiguity is produced automatically, and this ambiguity is experienced as an implicit pull on human intentionality to draw out the meaning of the dynamics encountered in experience. The more traditional and common-sense understanding of communication as an intentional meaning-negotiation is not inherently false in this view; however, it relies upon an emergence of structure via the inescapability of choices that are implied in the form of communicative utterances. Luhmann shares with other perspectives synthesized here the assumption that response is unavoidable, since all experience occurs within already partially-structured, partially-intelligible surroundings.

Thus the phenomenon of communication becomes “animated” and makes meaning within interaction via the generative power of ambiguities to prompt choice and action within pragmatic situated relations (Luhmann, 1992). It is not primarily the explicit utterances we make that structure our relationships, but the implicit and presuppositional dimension of our
communication that is central. In this view, the formulation of utterances matters, since each explicit formulation carries implicit and presuppositional force that relates and situates the utterance in existing structures of coordination. The content of a particular utterance calls for a response from other participants by virtue of its being formulated as such, and anything explicitly supposed can be directly negated as well as affirmed. This leads to various social and cultural norms that govern the strategic use of presupposition.

To position a communication utterance to presume certain things is to structurally privilege these presumptions since these presumptions are harder to disagree with. This is because the burden of explicitly uncovering and deconstructing underlying assumptions falls upon the other participant (Luhmann, 1992, p. 305). By indirectly articulating presumptions, the lack of response directly to them remains ambiguous, and as such responses that negate the content of an utterance can still be seen as implicit affirmation of the premises underlying it. It is in this sense that relational infrastructure can be seen as performed in communication.

**Communication as temporally constitutive.** Each of the above scholars also points to (at least by implication) an important temporal dimension to intelligibility that transcends spatial relations per se. The particular character of responsive structure is not defined simply by the addressees to whom we respond or to the ambiguities present in an implicit dimension of communicative acts, but also is particularly dependent on the temporal relationships among acts. We are responsive to the past and the future as well as to present otherness. This point is made by dialogic scholars drawing on Bakhtin’s work, particularly John Shotter (2003, 2006, 2008). Shotter characterizes the background within which communication occurs as a holistic “flow of mutually-responsive activity,” intelligible to subjects “as a set of openings and barriers to our actions – ‘given’ in relation to our present ‘position’ within ‘it’” (2003, p. 449; 2006, p. 594).
Shotter places the word *position* in single quotations to problematize the usage of the term. In particular, he argues that all parts of the whole are reciprocally implicated not just in terms of spatial position, but also across time and historical position as well. Because of this inexorable interdependence, transformations that occur in one position or region of the stream affect the whole, although these effects cannot be directly seen from the partial perspective situated within this stream. Affordances and constraints are temporal as well as spatial, in this view.

Shotter (2006) explains:

There is thus a distinctive ‘inner dynamic’ to living wholes not manifested in dead, mechanical assemblages, such that the earlier phases of the activity are indicative of at least the style of what is to come later — thus we can respond to their activities in an anticipatory fashion. In always giving rise to what we might call identity preserving changes, they and their ‘parts’ are always ‘on the way’ to becoming more than they already are. This is why their special, living nature cannot be captured in a timeless, ‘everythingpresent-together,’ spatial structure. Their special nature is known to us only in the distinctive ways in which they unfold in time. They thus require a special kind of understanding which takes their temporal ‘movement’ into account, a historical understanding, an understanding that entails, not an instantaneous ‘getting the picture,’ but an understanding that consists in an unfolding movement — like a piece of music — that has a unique temporal ‘shape’ or ‘identity’ to it. (p. 591)

In this view, then, there is an irreversible temporality that provides intelligibility to the coming-into-being of the world, and the temporal relations among actions and events are
constitutive aspects of coordinated behavior and what we come to achieve as stability.\(^9\)

Temporality is immanent in presence of objects and features that emerge in communication. Every utterance is oriented to previous actions that relate to it, as well as structured in anticipation of future possibilities implied in the utterance (Shotter, 2006). Shotter points to a duality within this temporal responsiveness: utterances respond implicitly to the immediately previous action or utterance, yet any given utterance should be concurrently seen as its own original site of calling, with its own set of presuppositions and implications, full of meaning potentials for future possible utterances or actions (Shotter, 2008; see also Lahteenmaki, 2004).

In some sense, all that has come before and all that may come in the future are potential addressees and constitutive aspects of a particular unit of communication. By virtue of its embeddedness within time as well as spatial relationships, an act of communication cannot stand outside of the particular moment in which it is invoked. This is implied in Smith’s notion of registration; a significant amount of work goes into making an object appear to be persistent and stable over time, and this is true all the way down to our most fundamental theorizations of how the world works.

**Reflexivity and redesign.** The above theorizations provide a basic notion of constitutive communication to guide the development of a metamodel system. However, based on the reflexivity of the metamodel, the rules that define how communication is constitutive must be capable of undergoing transformation themselves. As Suchman (1987b) points out, the

---

\(^9\) This resonates with the notion that the term *orchestration* best characterizes relational performance (e.g. Smith, 1996). Various literature in communication and organization theory suggests valuable lessons to be learned about creativity by studying communicational improvisation (e.g. Barrett, 1998; Weick, 1998). Drawing on metaphors of ongoing negotiation of fit via performer-group feedback loops, the notion of ongoing collaborative orchestration serves as a practical way to conceive the function of a metamodel system.
particular context of a situation forms the starting point from which the rules of interaction that effectively define that situation emerge. The affordances of a communicative environment are outcomes of both communication itself and the background conditions of an interaction. From a design perspective, this returns us to consideration of ongoing redesign. What role does redesign by users play in restructuring a situation based on a reflexive feedback loop that redefines the functioning of the metamodel itself?

Returning to the domain of computer systems, we can phrase the question in terms of what Gabriel and Goldman (2006) call autopoietic and allopoietic systems. In this context, the design question is how, exactly, could a reflexive system design entangle these two with one another. Such a system should operate allopoietically to model and constrain interaction according to the performativity of context and surroundings made up of various processes that run continuously. Users’ interfaces should not only allow them to act in relation to one another following these rules, but also enable autopoietic redesign that transforms the ways that the system constructs its allopoietic model. Interfaces for autopoietic redesign of the system are engaged from within the allopoietically-rendered environment where constitutive communication is itself modeled as an inexorable, yet modifiable set of processes and rules. Design occurs within embeddedness in these constitutive conditions and the system, as a result, might be described as the ongoing operation of restructuring a situation based on this reflexive feedback loop.

The open-ended participatory design and redesign are important aspects of the integrity of systems that aim to truly support embodied and situated interaction, a concern which places autopoietic systems at the forefront. A metamodel system must allow that the autopoietic mechanisms for reconfiguring the system be directly accessible to users, yet conditioned in some
way by the allopoietic coordination mechanisms that provide the system’s model of embeddedness and interdependence. This reflexivity is at the core of the problem we have in defining clear boundaries for the system. The approach taken here is to define the entire system as including both the allopoietic and autopoietic dimensions within the software and entangled with one another.

**Designing the “Hmmm…” system**

**Clarification of the design boundaries.** The theoretical considerations above outline a particular design domain. With this definition of the problem in mind, I have attempted to design an interaction environment that is capable of modeling the complex and reflexive relationship of communication to the design of a situation. The intention is to design at the level of infrastructure for ongoing redesign, rather than solely on the allopoietic mechanisms for modeling complexity.

Now I will return to the question that opened this thesis: How to design a system that reflects the perspective outlined above. Can a metamodel approach produce a system that enables communication and design to be applied reflexively to the interface while still enabling the first-order interactions themselves?

To address this I undertook a long-term design project to develop an integrated system. The process was one of synthetic thinking, whereby I would attempt to move from theories of communication and entanglement, reflexivity, etc. to strategies for modeling and simulating these conditions within a communication-mediating interface. Through multiple design iterations combining different theoretical constructs, I came to a stable set of underlying ideas that form the core of a collaboration and redesign framework, which I have named “Hmmm...”.
From here forward, I will refer to the proposed system as “Hmmm…” in order to clarify when I am making reference to it.

This initial design was not undertaken with knowledge of systematic approaches to designing computational architecture. Rather, I took a general knowledge of contemporary technological affordances and speculated about how to develop large-scale systems modeling entanglement, complexity, flux, and the constitutive nature of interaction and communication. The design and prototyping process did not involve considerations of particular programming languages or constructs, software architecture, or computational deployment of the system in general. These considerations are more concretely addressed as the design progresses, and this will be demonstrated in the next chapter through an analysis of existing collaboration frameworks.

**Constructing a constitutive metamodel architecture.** Initial design efforts focused on defining an interaction architecture that provides basic rules of interaction, which create a particular first-order model of communication-as-constitutive. The resulting structures provide rules for system-intelligible action and the resolution of various combinations of communicative events within the interaction space. The model underlining this communication architecture was then reflexively applied as a metamodel to guide the development of mechanisms for ongoing redesign of the system itself. In particular, the proposed solution for “Hmmm…” was to model each user, the system itself, and the shared situation, all as objects within the interaction episode that are open to modification. By treating the system and the intersubjective results of its rendering as participants, the intent is to allow users to intervene in system functionality directly, without categorically separating these autopoietic dimensions (e.g. in system options or settings menus) from the interaction itself.
In the remainder of this chapter, I will outline the proposed system, focusing attention on defining the structure, at the expense of describing what the interaction itself might look like within the system. While this approach ignores concrete users and use cases, it provides the foundation for considering whether the proposed system is feasible as collaboration technology. Due to space considerations, considerations of specific use cases and related issues will be left for future work.

The system has various structural components. I will begin below by providing basic definitions to guide subsequent discussion of the proposed system. Next, I will discuss the construct of meta as a way to represent layers that provide depth to elements and relate them to one another. In the following sections I will discuss particular meta dimensions of the layered architecture proposed. I will begin by separating descriptive and functional as two complementary modes that provide a duality to the structure as it is instantiated. Next, I will discuss particular structures of registration and temporality as providing further layers of depth to representations of interaction. I will briefly consider exteriority as an additional meta dimension that operationalizes the uncertainty present in all representations. I will then discuss a proposed way to dynamically organize the appearance of elements in a limited perceptual space via salience and persistence attributes. Next, I will describe generally how the relationality among elements is rendered over time, defining structures of responsiveness between elements within interaction and providing the basic language for end-user design from within “Hmmm…”. Finally, I will describe the proposed way of instantiating users, the system itself, and the situation itself as reflexive meta-objects.

**Defining the system.** The “Hmmm…” system’s basic allopoietic function is the recording, processing, and feeding back of human inputs in real-time via a specified interface
and visualization language. In order to minimize confusion, I will refer to each act of system-
intelligible input and each unit of user-intelligible system execution as an event and refer to each
rendered unit of meaning as an element. What counts as an element in the operation of the
system is not entirely predefined, but rather depends on the parameters set for how the system
will render and relate events together. For simplicity’s sake, it could be argued that any event or
defined set of events can potentially be rendered as an element. Particular types of events and
sequences will be referred to as actions; these can be thought of as elements possessing
functionality that can be instantiated to directly affect other elements over time. Since rendered
elements have an internal architecture in the proposed model, I also speak of the parts or aspects
of an element. Of course, each of these parts can operate as elements in themselves, and thus,
when I refer to elements this should be taken to recursively include all possible elements present
within the full architecture. When I am specifically referring to something in terms of its
relationship to a larger or more salient element, I will use the term aspect to distinguish.

The user interface. As defined here, the “Hmmm…” system is not tied to a specific
interface or particular parameters for interaction, but rather at the level of defining the general
dynamics and architectures afforded to designers and users. Very different interfaces and
implementations could conceivably provide the same functionality, though each would afford
different possibilities for experience and thus lead to different interactions. Within the design
and prototyping process undertaken here, a multitouch surface was used as an example interface
to provide a more concrete vision of what the abstract ideas might begin to look like in practice.

Theoretically, the “Hmmm…” system operates by recognizing and processing users'
actions over time and displays various renderings of action as interactive elements on the input
surface according to the architecture specified below. Users act to reference a particular element,
aspect, or group thereof in order to interact with these elements. Users affect the continuous rendering of the visual space whenever they click on some displayed element and perform some action, and these very actions that reconfigure visual space are themselves represented as interactive event elements. The rules that govern the function of the system, as well as the actions and events that are executed by the system, are represented similarly and thus can be observed and modified from within this interface as elements. The system passively records all clicks and actions as a stream of events, and continually processes these to rerender the interaction space.

The “Hmmm…” approach is to define an aspect and relationship architecture at the outset, and to have events and their sequences automatically populate various elements and aspects. This occurs via rules and logics for punctuating a flow of action over time, and is further specified concretely based on parameters and additional layers of functionality that can be invoked for particular use situations. Layers of functionality and configurations of parameters could be encapsulated as plug-ins built up over time from the ongoing codesign of the “Hmmm…” system. Below, I will describe the basic architecture of aspects and a general model of interaction dynamics.

**Metalevel architecture.** The system calls for a way to render a truly diverse multiplicity of elements and layers, each with uncertain and radically multiple possible relations to the others. I chose to model this characteristic through the architectural specification of *meta* aspects. This was initially designed as a way for an element to refer internally to something specifically larger than it, e.g. the situation or field to which the element relates. Through the design process, I have come to realize that the meta specification encompasses more than this; it is implementation
of aspects themselves, providing a language and overarching specification of how layers and aspects can relate to elements and to one another.

Meta is an infrastructure of aspects that are automatically and manually populated in relation to a given element. By default, all elements possess a theoretically infinite multiplicity of meta dimensions, and will begin with a few specific ones making up the proposed meta architecture, outlined below. A surface or presentation layer that is directly interactive and defines the visualization of the element, is treated as separate from all other meta layers representing other elements in relation to it.

All meta layers attached to a given element may contain local information and functionality, but operate largely through reference to (explicit or implicit) external elements. In this way, meta layers are semiautonomous elements situated within any given element under consideration. This is because meta layers are composed of the same recursive elements as the rest of the system, albeit differently situated. Furthermore, the system should accommodate multiple, potentially incommensurable dimensions and interpretations of meta via replication and concurrency strategies.\textsuperscript{10} The intent is for a basic structure to provide flexibility in interpretation and use, allowing users to populate aspects of the architecture according to whatever interpretations are afforded by the use situation and interface.

**Descriptive and functional dimensions.** The “Hmmm…” system makes a distinction between functional and descriptive action as a basic way to structure utterances and to build meaning. This separation intends to define a duality within any given element’s logical

\textsuperscript{10} Mechanisms for replicating entire run-time environments may require massive computing resources and remain unfeasible for most situations, however the basic potential to do this remains a key design goal.
structure, partially separating the element’s underlying behavior from its visual appearance. In short, elements are populated along both functional and descriptive dimensions. Each element and its aspects perform actions and have functional behaviors and effects relative to the space of interaction, coupled with descriptions by which they are referred, explained, and linguistically mediated. Within the flow of time, automatic defaults and manual controls define the oscillation between functional and descriptive modes for interpreting inputs, defining whether system will parse any particular events as descriptive or functional content. As can be seen in other areas of the architecture, the separation is intended to be only partial, with some degree of entanglement operationally defined between them.

This distinction is reflexively applied to the functionality of the system, allowing for redesign to be performed from within. This serves to create a practical, reflexive interface to subjects and the situation as a whole within the very structure of the rendering system. Descriptive and functional dimensions have a two-way relationship, and users can specify whether and how they want to make some purely descriptive utterance effective in its conditioning of interaction (e.g. where they wish to situate it, whether to relate it to some functionality, etc.). This proposed distinction thus serves as a strategy for flexibly separating and recombining pragmatic and descriptive dimensions of utterances within interaction.

**Registration as a model for entities.** I use Smith’s (1996) theorization of registration and stabilization as one basis for the ontology of elements within the system. Registration is an overarching metaphor and conceptual specification for how objects are created and come to interact with one another. The basic registration that the system models involves a three-part relational unit consisting of (a) subject, (b) object, and (c) the act of registration or registration link. These are proposed to have a dual nature here, where the subject, object, and registration
link count as meta aspects within a single element, yet are also contained as three separate elements in relation to one another.

In elements that persist, multiple registrations will often be integrated into a single focal element as further populating these aspects. Because of the ubiquity of registration, the system needs a way to simplify this complex rendering in most cases. The registration architecture can be defined by a set of code that runs upon rendering an element whereby the three part structure is unpacked and repackaged into elements and aspects, ordered depending on parameters for rendering. For example, the object aspect of a given element is often functionally treated as its whole form, while the registration link aspect can be taken as an element of relationality that corresponds to a given set of objects and subjects and their relationships. The same sort of unpacking and repackaging process applies to the rendering of other aspects considered below. In this way, an entire architecture can reside as aspects upon the rendered element referencing other, functionally distinct elements that populate its aspects. As suggested above, this general functionality is open to various extensions, whereby objects can act as subjects, for example.

**Temporal architecture of entities.** In addition to being expressed in a registration unit, the existence of any given element is connected with its temporal relationships to other elements and referenced to a synchronizing timebase. As a starting point, I propose that elements have an internal temporal architecture defined by a three-part aspect structure consisting of (a) past, (b) future, and (c) the present. The temporal architecture provides for some of the possible richness of an element that always has a history encompassing relationships with previous elements as well as a future encompassing anticipatory consequences and meanings. This temporal architecture is crucial for enabling the persistence of a shared referent across multiple
registrations, and serves to signify path-dependence within elements, marking the situated contingency of an element’s coming-into-being.

The aspect called present is simply a way to conceptualize the surface of an element in its current manifestation, including active processes and responsive potentials and interfaces. This is the site whereby the current state of the element is defined by the system, and where the appearance of the object is configured and functionality is executed. Events, by default, can be treated as directed toward, and rendered into, the present aspect.\textsuperscript{11}

As a particular element is responded to in interaction, the past aspect of this element is automatically (and manually) populated with a representation of the events that constitute its current state at the moment it is referenced. Within the past aspect of a given element, by default, the system will record previous elements and events that have been tied to the current form in various ways. The future aspect is a site for functional data governing the anticipatory behavior of a given element.\textsuperscript{12} Descriptively, the future might contain information about the purposes, plan or goals, of an element. Functionally, the future aspect can be used to specify the element’s potentiality and acts as the site from which responsive functionality is activated.

With this temporal architecture, the presence of an element changes over time automatically. As an element persists, the past aspect of the element defines the constitutive events and relationships as traces of previous states. The future aspect can similarly contain intelligible traces of anticipated future states. Combining this aspect architecture with a

\textsuperscript{11} Even actions that directly specify a response to history or future dimensions will themselves occur automatically on the present dimension where they are first encoded as structural effects upon interaction.

\textsuperscript{12} Functionally, the future aspect will generate anticipatory responsive calls that, along with other meta dimensions, govern the integration of the object into the situation.
mechanism for synchronizing events and processes to a timebase, an element can be structured with internal temporal relations that govern its behavior in relation to other elements, and elements and their relations become meaningful in historical context.

**Exteriority and indeterminacy.** An additional meta dimension representing that which is unknown, unknowable, and generally exterior to a situation is included as a meta aspect in the “Hmmm…” system. Encompassing the ambiguity that emerges from intersubjectivity, and the types of uncertainty that emerge from complex interdependence (e.g. Nicolis & Prigogine, 1989), it is practically important to acknowledge that the shared rendering of the situation will invariably leave out or make invisible some presences that play a material role in constituting the situation. The meta dimension for exteriority is a way to symbolically keep this inherent unknowability present within all interaction. This dimension was also imagined to be a module for the application of random processes and external calls that problematize the ontological security of any given element.

**Salience and persistence.** As a particular element becomes programmed or rendered with explicit and implicit content or relationships, it is visualized according to dynamic attributes of salience and persistence. These attributes govern the relative power of an element to affect other elements in the space, prescribing, to a degree, the order of influence and the hierarchy of resolution. A set of rules defining how these attributes function is part of the basic functionality of the system, and these would be modifiable as parameters for various situations. To do this, “Hmmm…” implements an additional layer of constitutive visualization and relational dynamics upon the base rendering activities of the system. This additional layer defines the relative importance and solidity of elements and aspects within a semi-shared visualization of a situation.
Salience is a construct that seeks to model the urgency, immediacy, and general probability to evoke attention, of a given element relative to others. This is done primarily by restructuring the appearance of the element over time. In particular, an economy of salience would be expressed by variable amplitudes, rates, and timbres of visual behavior of elements.

Persistence seeks to model the solidity, permanence, or continuity of a given element. It can also be interpreted and extended to express the degree and type of reification or the general materialities of an element and its referents. The embedded elements that are outcomes of interaction must be able to grow and become persistent, relatively stable or repeated boundaries constructing meaningful objects and subjects in relation to an environment.

According to the responsive structure of relations within interaction, elements become less prone to disappear or become destabilized as they become more structurally embedded, that is, as they are enrolled directly and indirectly in numerous registrations. Thus, salience and persistence can perhaps be thought of as meta aspects governing effective priority, functional integrity (or robustness), and general orders of appearance. This formulation requires that new elements are capable of building upon and within existing objects based on their salience and persistence, such that structures can become embedded within the surroundings based on their implicit relevance to the situation at hand.

**Relationality among elements.** Default forms of relation are particular ways to constitute elements in relation to one another. These are expressed as separate actions running code upon the elements invoked, and as representations of the effects of elements upon one another. In the broadest sense, elements of relationality can be seen as extensions of the registration link specifying particular effects of various registrations across time.
Elements can be actively related together in order to build structures of meaning within interaction. This happens automatically depending on parameters, whenever responses to actions occur implicitly and explicitly. In various ways the responsive logic of the system prompts users to articulate the functional relationships between elements within interaction as it happens. Forms of responsive restructuring include direct and indirect response to an element as such; direct response extends to various actions that can transform an element, such as relating two elements along a particular dimension, affirming or negating a particular element or action, and grouping or dividing elements. The intent is to provide flexible affordances for relating elements to one another to build structure and achieve particular goals within interaction.

**Responsive structure of registrations.** Any element becomes interactive and embedded upon being punctuated, resolved and rendered, interdependent with all other participants registered by the “Hmmm…” system as unique subjects.\(^{13}\) This involves a method for system-intelligible punctuation via *starts* and *finishes*. Both user actions and system actions are visualized with this start-finish logic, enabling intervention into running processes. An event that finishes a given articulation is the basic means of decisively acting upon some element, noting that the system has defaults for creating automatic finishes to various processes that are not explicitly finished by users within a given window of time. Before the functionality of an articulation is automatically resolved, however, the system may ask users to respond to the articulation in particular ways, in order to clarify its relationship to existing elements and to accept the changes that will result. During this time the element itself is treated as being

\(^{13}\) This is a simplification, since the *resolution* of a particular articulation in a particular way does not occur immediately, but varies depending upon the default parameters for the type of action and the relational effects that it generates.
suspended, during which time it can be responded to in various ways but does not perform the effects of its rendering.

The “Hmmm…” system includes an action for interrupting a particular process that is underway (started but not finished), so that users can interrupt a given process in order to suspend and modify it, or to override it with a new process. The interrupt action pauses and suspends the current process (whether a user sequence or a system rendering sequence) and begins a new sequence that takes priority over the suspended process. The function of an interruption will resolve before the target that it interrupted, and thus can potentially negate or otherwise undermine any given rendering.

The punctuation that defines the utterance is based on a particular finish command, akin to pressing Enter, which creates a registration unit and puts this unit “in play” within the shared space. As mentioned above, relations are stackable and the system provides a means of developing an object over time before submitting it to the system as an intentionally-punctuated articulation. The metaphor of making a wager in the game of poker provides a starting point for understanding the responsive logic modeled in this prototype. Any punctuated articulation will be treated as a wager on reality and will be passed among other subjects in calling for some response. As mentioned above, the same logic covers system-generated objects as well as clarifications requested by the system to a user-generated object. In any case, punctuated articulations are presented as a call to users (and other objects) to affirm, negate, or transform it before it executes. Of course, the structure proposed here is much more open than a poker game by default, insofar as there is no strict ordering to response and there is no absolute requirement of direct response in terms of agreement or disagreement.
This logic of process, punctuation and interruption creates a real-time temporal relation among process elements and events, and establishes a real-time economy of attention and dynamics to the conversation. If we apply the effects of the responsive logic to the attributes of elements such as salience and persistence, we can develop communication environments where the ontology of objects and structure can be powerfully conditioned by the particular form of interaction taking place in real-time. There are important constraints of this openness, however, as the system will always functionally render some relationship between objects beyond that which is explicitly specified, and thus additional intervention may be required to modify, suspend, or displace some of the functional relationality that occurs between objects by default. Since you can also suspend and subsequently negate system actions, the interrupt functionality combined with other actions provide all sorts of ways to create, as well as to address, critical errors and breakdowns in the system. For the system to remain robust requires a strategy for workarounds, which might include complex mechanisms for virtualization that allow suspensions of functionality in particular sectors or aspects of the system.

This highlights the need for ways of creating and managing multiple orders of rendering to enable the creation of subobjects within larger articulation processes. The proposed strategy can be seen as an implementation involving multiple resolution logics in play simultaneously, such that a user can build according to one logic and then submit the results within a second resolution logic. This is akin to having a mechanism for internally punctuating complex objects while still within the process of finishing a single articulation. In the simple example of differentiating between personal and social workspaces, the effect of punctuation mechanism within the personal space is not to render something as responsive to other users, but to resolve internal structure to the object and relational definitions. Thus users can build by drawing upon
the logic of the system before submitting the product of this construction into the shared registration space for response by others and resolution with larger situations.

**Managing complexity.** Many of the above suggestions for functionality point to incredible problems of complexity and reflexivity in design, but also presumably in managing the operations of a running system. Indeed, to model interaction in terms of this recursive architecture, some selectivity even within the particular mechanisms proposed here is a practical limit on both the system and users. Much of the complexity of structure possible within the system’s parameters for articulation will have to remain implicit within a real-time human interaction episode. How can the system and users manage this complexity to create a relatively stable and actionable space of affordances and constraints? The intent to is develop methods for the system works together with users to manage complexity and resource allocation based on situational needs, however no particular solution was included in the current prototype. Rather, in the simplified prototype the system defaults to a minimal construction of element architecture with the potential for users to expand the complexity of these structures upon directly selecting any given element. The deeper structure is implied but only rendered and brought into effective functioning as a user clicks down a particular path, creating more detail.

**Reflexivity as a design principle.** The above indirectly addresses techniques for reflexivity that allow redesign of the engine *in situ*. Seeing the rendered environment as something more than a container for interaction requires that we consider the situation as a dynamic entity in its own right, equally capable of being transformed by other participants within interaction as it is capable of transforming these same others. “Hmmm…” must therefore not only be designed to model the constitutive communication of its users, but also it must accommodate a reflexive dimension whereby the system, users and situation themselves are
somehow constituted in interaction. This is initially done by performing an *opening registration* sequence that constitutes the bootup process for a particular instantiation of this system. This opening registration defines initial articulations of the situation as such and the entities that presumably register one another, establishing the grounds for further interaction. The particular strategy chosen here is for the initial registration to render directly-actionable elements that correspond to the system as a subject, and the situation as a metaobject, as well as subject-objects representing each user. In general, the intent is that by interacting directly with these constructs and their corresponding aspect architecture, users can cross the boundary that separates what happens within a situation from the background that makes it possible.

The “Hmmm…” system provides a representation of the rendering system itself as an interactive participant in the very relationship-building that it renders. Programmed to express itself in ways that mirror human user actions, the system subject-object construct can be responsively embedded in real-time relationships with other subjects who are capable of transforming (suspending, overriding, reconfiguring) the system’s core functioning. The system subject-object is modeled on the structure of user subject-object, and thus its performance could be represented as an inverse of each user-driven registration, and multiple system subject-objects could be developed to correspond to modular functionality. In any case, the system subject-object can generally be said to allow users to reciprocally observe and transform the system’s subjectivity as it constructs and coordinates elements.

In the default mode of operation, users can actively restructure the workings of the system by responding directly to the objects created by, and constitutive of, the functioning of the system itself. The system processes responses to its own actions in order to modify the effects of these registrations relative to the situation. However it seems impossible to design a
single system subject-object that is capable of reflecting its own internal processing and simultaneously process responses that attempt to transform its own constitutive processes all the way down. In a sense, then, the system subject-object is an interface of abstraction to the underlying processing that takes place. The particular levels at which the system’s actions are available as objects must be based on some default reflexivity parameters applied in any given instantiation of the system.

User subject-object constructs are persistent representations of the system users and their relations to other elements within a situation. These constructs define the history and potential affordances of any given user, alongside any other descriptive content and functional constraints arising from within the use situation. The user’s interface to the system could be merged with, or encapsulated in, the user subject-object construct; this was the case with the initial prototypes.

The situation object is a single central interface to the particular situation being constituted by the “Hmmm…” system and its users. It is akin to having an element that refers to “this situation itself” that is reflexively tied to what goes in within the situation but also actionable as an element participating in the situation. Functionally, this object defines the currently default parameters of engagement, as well as containing a record of the situation’s intersubjective content.

Via direct interaction with the situation object and its layers, users can change and expand the meaning and functionality of the situation as a whole without encoding these changes directly into the system-subject. In fact, the intent is that the situation object might be able to implement global constraints on the instantiated system and users based on the responsive logics that the situation itself may be subject to. At this level, the system can support a practical,
structurally-significant engagement with the world beyond itself that is not completely reliant on the interpretations of the particular users recognized as part of the situation, since the situation object could persist and be perhaps affected by others outside of the local situation.

The System Decomposed into Modules

For the sake of brevity, I summarized the system and left much of the proposed system relatively unexplored. There are many elements of the proposed system, and so far only a surface-level consideration of implementation has been included. In general, the above attempt at description betrays naïveté on the part of the researcher in articulating the structure and mechanisms of an interaction support system. However, the basic concept of the system has been stated. To bring a clearer focus to conversations around feasibility of implementation, I will summarize the above with an architectural description of the “Hmmm…” system, separating out functional subsystems defined in terms of computer-mediated collaboration support technologies.

First, we must define the proposed system in computational terms, in order to relate it to existing software systems that serve similar purposes. Following Edwards (1995), the “Hmmm…. system could be categorized as a collaboration support environment. Such systems, according to Edwards, consist of one or more collaborative applications supplemented by a common, high-level application programming interface (API) and runtime support mechanisms that allow developers control over the behavior of applications during execution. Taking into account that the actual collaborative applications are included in this definition, I will operationally define the “Hmmm…” system as a collaboration support environment for the purposes of summarizing its functionality and approaching questions of feasibility.
As a standard collaborative application, “Hmmm…” requires basic mechanisms for supporting interaction among multiple participants in real-time. This would consist of a set of foundational mechanisms, a base rendering system that constitutes the background conditions of interaction and is involved in generating the elements of interaction, providing the computational infrastructure for the real-time distributed environment. This foundation includes all the basic requirements of a collaborative application: mechanisms for creating, modifying, visualizing, distributing, and synchronizing a multiplicity of structured yet open-ended computational objects via some end-user input methods. This encompasses the ontological relationships among constructs as well, the experienced interface of user-intelligible affordances.

As a collaboration support environment, “Hmmm…” requires mechanisms for redesign and transformation at runtime, namely, a common API and runtime facility. In the case of the proposed system, the ability to modify the runtime execution of the software is shifted from the responsibility of developers directly to the users. Also, one of the implications of the particular assumptions outlined in the previous chapters is that the features of the collaboration support environment (e.g. common APIs and runtime facilities) are in some sense entangled with the features that provide its function as a collaborative application. That is to say, in the case of “Hmmm…” there is a reflexive relationship between elements of the collaborative application and elements of the API and other runtime-intervention facilities. In a sense, this proposed user-responsive real-time reflexivity may represent the most unique aspect of the “Hmmm…” approach.

The architecture of any collaboration support environment will embody certain assumptions about communication and about some ontology of elements and their relations, and the particular implementations of functionality provide the unique identity of any given system.
Some proposed features, components, or subsystems within the “Hmmm….” design converge, it is hoped, to form a different sort of collaboration system foundation, one that is in line with various constitutive theories of communication as outlined in the preceding chapters. Below, I will consider particular features of “Hmmm….” that embody it’s unique identity as a constitutive metamodel technology, ignoring for now a further specification of the general requirements for collaborative applications and support environments.

Generally speaking, a transmission model of communication might lead to systems that treat the background environment upon which interaction occurs as static and only activated when human users initiate action. By contrast, the “Hmmm….” system models the background as internally dynamic and continuously changing, punctuated intelligibly not only by explicit user actions but also by incidental user behavior fed back and attributed meaning by the system, via automatic punctuation of time and space relations. This encoding of a particular temporal structure of reality, an ongoing flow or stream of recording, and mechanisms for punctuation of this stream as the fundamental way that elements are constituted, is one of the unique requirements of the “Hmmm….” system. Furthermore, it is important to the proposed design that these temporal mechanisms operating at a relatively low level, “listening to” and processing the environment in terms of small increments of time from a human perspective. This particular feature defines a type of process ontology, and would theoretically be active even within a single rather than multiuser instantiation of “Hmmm….”, in order to create multiplicity and interaction.

Another requirement of collaborative applications is that they provide support for multiple participants in interaction. All collaborative applications need some mechanisms for synchronization and distribution among various locations (e.g. clients), in order to create a semi-shared models of a dataspace that is an outcome, and constraint, of user interaction. Because the
constitutive view taken here problematizes the notion of a single, transparent, shared reality, the
“Hmmm…” approach calls for a user-driven model of the rendering and resolution of
multiplicity, as complexly-distributed and semisynchronized intersubjective realities. Rather
than aiming for simple and transparent distribution and synchronization of objects, the
“Hmmm…” application would model particular notions of ongoing dynamic responsiveness and
rendering logics that go beyond simple locking of objects and reflect the complexity of
interaction upon, around, and with a multidimensional object.

As stated above, the features of a collaborative application and those of a collaboration
support environment are entangled within the “Hmmm…” approach. Any collaborative
application requires a way for users to create, modify and relate together elements along with an
ontology of elements and relationships available. The “Hmmm…” approach adds atop this a
requirement that the underlying structures and affordances for interaction be reflective and open
to redesign and modification at runtime. In particular, the language for building and relating
elements together functionally must also encompass a reflective capacity for intervening directly
in the functioning of the system. Thus the system includes an ability to modify the structure of
underlying realities (the reflection aspect of building), and the particular architecture that is
provided as part of the partially autopopulated ontology of objects. This includes the temporal
buildability, as well as the other aspects of how meta dimensions operate.

In general, what is being added to a standard collaborative application is a process
ontology simulation of a particular sort, encompassing a complex version of intersubjectivity that
complexifies the appearance, distribution and synchronization of objects. What is added to a
collaboration support environment is an approach to build reflexivity throughout the architecture
that includes mapping the rendering methods and default relational ontology to metaobjects
representing the system and situation, as well as users. Together, these features of “Hmmm…” aim to open up an ecological space for design and communication to take place constitutively, via reflection-in-interaction making system explicitly redesignable from within at a direct level. The rest of this thesis will be concerned with asking whether existing technologies afford a constitutive metamodel framework already, and what can be learned about development of such a system by examining existing attempts to address the problem of entanglement in situated design and communication.

**Conclusion and Statement of Research Questions**

In choosing research questions, I take the route of practical application of the model above within realistic constraints. While there are many ways to inquire about practical applications of these ideas, I focus here (for reasons described in the next chapter) on the general technological (rather than abstractly computational) feasibility. Whether the mechanism is computationally feasible is too broad (and perhaps formal) of a question to answer here, and outside of my area of expertise. I will ask a more limited, more immediately practical question: could existing technologies provide the desired functionality and infrastructure for “Hmmm…”? Considering whether existing systems and frameworks can support an implementation of the proposed model will help us understand whether the proposed system is feasible, and inform any subsequent development by surfacing issues, approaches, and relevant technologies. In the following chapter, I address this research question by analyzing current frameworks and environments for communication and design that may offer some answer to the immediate feasibility of the “Hmmm…” approach.
Chapter Three

Introduction

Are existing interaction frameworks feasible as sites for understanding and developing the “Hmmm…” prototype system? In this chapter I seek to explore the affordances of existing collaboration technologies for comprising a communication-as-constitutive reflexive metamodel. First I will clarify the scope of analysis and describe the methods used to identify relevant technologies. Next, I will provide a rationale for selecting the specific technologies under consideration, briefly outlining the broader solution space as it was understood via the research process. Subsequently, I consider two selected technologies in depth, drawing out affordances and constraints that inform the feasibility of a “Hmmm…”-like system. I conclude this chapter by attempting to directly apply these findings back to the “Hmmm…” system by way of the research questions.

Research Scope and Methods

In the previous two chapters I outlined some of the complexity underlying the design of dynamic communication environments, and made a case for a system that attempts to model some of this complexity as a means for developing a richer and more open-ended communication and design infrastructure. The articulation of a solution via a set of interacting features comprising the “Hmmm…” system leaves many questions unanswered. In particular, questions of implementation, and of the specific formulation of structures and their relations, are not
directly considered within the previous chapter, and are only hinted at within the actual demonstration prototype developed from this model.\textsuperscript{14}

Design work takes place in an environment, and the way that designers understand the problem space they are addressing influences the forms taken by the outcomes of design. In order to assess whether “Hmmm…” might conceivably satisfy the requirements of emergent, reflexive communication and design work, an attempt to discover other solutions addressing a similar problem space is undertaken. Technologies that address the problem space of coordination among distributed users within an open infrastructural framework that accommodates flexible redesign represent a set of data, which can be analyzed to reflect on the “Hmmm…” design, it’s theoretical foundations, and the practical possibilities for extending a reflexive, constitutive metamodel as an approach to participatory infrastructure design.

The proposed system is merely a conceptual system, a thought experiment, at this point. It does not contain a specification of underlying architecture that is fully developed, and could be directly mapped into code. In order to assess the feasibility, in practice, of these ideas, I am checking the system against real-world technologies that articulate a similar problem space, with the intent that such analysis will yield findings about whether the proposed approach is actually feasible to implement. The goal is to enable critical self-assessment as to whether, on the basis of previous work, the unique features comprising the “Hmmm…” approach are actually feasible

\textsuperscript{14} A prototyping process to implement a Wizard-of-Oz style simulation of “Hmmm…” was undertaken, resulting in a workable, though far-from-ideal, offline demonstration. Informal demonstrations generated enthusiasm and a lot of creative speculation on use situations and extensions of the framework. However, as a research approach, testing the prototype was not feasible due to the complexity of rendering, and the need to develop an entire use scenario with preexisting content in order to create immediate end-user intelligibility of a metaframework.
to develop and represent substantive design contributions in the area of collaboration support technology.

Despite the increasing ability of the researcher to specify elements of the proposed system in computational terms as the design process progressed, the depth of technical knowledge required to directly answer the question of feasibility with a computational proof-of-concept is beyond the expertise of this researcher and beyond the scope of this thesis to fully investigate. Given this limitation, designing an implementation from scratch is also not possible for the research under current conditions. The clear option appears to be to outline a feasible implementation by identifying and recombining existing technologies to meet the requirements of the proposed system.

Across the entire spectrum of applications supporting collaboration in one or another form, there are undoubtedly numerous systems that capture the spirit and affordances of one or more components and subsystems from the proposed “Hmmm…” architecture. The same could be said for the broad field of technologies supporting design and development of applications by end-users. However, the researcher lacks the preexisting knowledge of the field of relevant technologies, and defined in this way, such a field is far too broad for the scope of this research project. Furthermore, the researcher lacks sufficient technical knowledge to understand how a variety of unique and diverse systems can be integrated together to provide a total implementation of “Hmmm…” features.

With these limitations, it will be difficult to draw conclusions of feasibility via consideration of a wide variety of systems that capture only individual components or subsystems of “Hmmm…” in isolation. Instead, I’ll take a practical approach to focus on
already integrated collaboration support environments. Are there already collaboration support systems in existence that would accommodate most or all of the major elements of the proposed system? In other words, are there frameworks already in use that could theoretically support a model of interaction such as this, and how do these existing projects inform the ongoing design of “Hmmm…” system?

However, I do not attempt to answer the question of feasibility of actual widespread development and adoption of a “Hmmm…” framework. In other words, I have constrained this study to the general architectural possibilities for imagining “Hmmm…” as a real system. Therefore, the selection method does not give preference to the most widely supported technological platforms and programming languages within professional expert communities, despite the notion that these might arguably be the most likely actual constraints on the development of an actual “Hmmm…” infrastructure. Instead, I sought to discover conceptual design of frameworks for collaboration that were developed alongside actual computational forms. I required that any system under consideration here be established at least as a prototypical proof-of-concept, even if the system or its underlying technologies are not currently in mainstream use. Though not including mainstream social or cultural acceptance of the tool or its underlying implementation platform as criteria, I sought to ensure that any given system under consideration was embodied in a functional prototype, and that any candidate approaches were relatively well-cited in the literature and clearly represented a sustained inquiry into the problem space.

In this framing of the research questions and the approach chosen to address them, there is still a degree of separation of the designed artifact from its underlying implementation, however this separation is not categorical to the inquiry. In particular, considerations of
implementation are included when assessing the affordances for redesign that any given system provides. In most cases, of course, the affordances that knowledgeable users see and aspects of the underlying implementation are entangled to at least some degree, and therefore implementation is important for the study-at-hand insofar as it is tied to the general communication and design affordances as they are experienced by the developer and knowledgeable user. While this approach may not lead directly to any conclusions about best real-world implementations of the “Hmmm…” system, it provides a way to think more concretely through requirements of such a system, and for drawing on existing design expertise within software architecture and systems design. Thus at the outset of the inquiry, the questions asked were: what sorts of open-ended multiuser collaboration infrastructures and frameworks have been implemented as actual communication and design environments? Furthermore, what can we learn from these existing software projects about the feasibility of the proposed system? What conclusions can be drawn from an analysis of existing collaboration support frameworks about the possibilities for modeling a constitutive perspective on communication and design along the lines described in the previous chapters?

An operational definition of feasibility in the context of this study is grounded in the possibility of developing implementations of the “Hmmm…” conceptual architecture via existing technology with rich affordances in critical areas. What I will seek to assess is the feasibility within existing collaboration frameworks for implementing this type of constitutive metamodel approach at the level of basic user-intelligible constructs that are supported by computational implementations. I define this level as the human-intelligible logical construction of the system at the interface, constructs that are often but not always shared among developers
and knowledgeable users, and taken to include implicit as well as explicit dimensions of the interface. This is discussed more below within the section on filtering the dataset.

**Procedure.** In order to answer the research questions outlined above, I undertook to review a broad set of collaboration support environments, selecting out and subsequently analyzing approaches that seem to offer comparable solutions in a similar problem space to “Hmmm…” To discover existing collaboration frameworks within the problem space, I first undertook a broad review of technology in the areas of CSCW, groupware and collaboration environments. I began my data gathering by reviewing abstracts from the proceedings of recent conferences on CSCW, groupware, and collaboration environments. I also used other organic search methods; I performed searches of keywords and tags on Google Scholar (http://scholar.google.com), as well as link aggregation sites such as Delicious (http://www.delicious.com) and Digg (http://digg.com). Furthermore, I identified and reviewed various blogs and websites devoted to collaboration technology. In each case, I reviewed the results and did further inquiries into potentially relevant systems. I reviewed journal and conference articles and abstracts, project homepages, introductory documents, documentation, and attempted to identify current states of development and user communities.

---

15 Proceedings reviewed included the following: CSCW 2007-2010, C5 2005-2009, Group 2007-2010, BCS HCI 2008-2010, ECSCW 2005-2009, DIS 2008 & 2010, NordCHI 2008 & 2010. These are representative sources of research within the problem space, however as a whole they demonstrate a bias toward CSCW/HCI and a Western (U.S. and Western Europe) bias.

16 In particular, search terms included basic combinations and variations of the word “collaboration” combined with words such as “technology”, “software”, “support”, “environment”, “framework”, “real-time”, “reflexive”, “open-ended” and others including either replacing or combining with the term collaboration, including “design”, “communication”, “CSCW”, “EUD”, etc.

17 Some examples include Collaboration 2.0 blog on ZDNet (http://www.zdnet.com/blog/collaboration), ReadWriteWeb (http://www.readwriteweb.com/), Mashable (http://mashable.com/), Download Squad, etc.
As I began to get a clearer sense of the various projects and connections among them, I did increasingly specific searches centered on further details of relevant systems and approaches. As I was reviewing this data, I sought to map the evolution of a particular system through various design iterations under different names, which was sometimes difficult. In many cases, I found it necessary to conduct searches based on specific authors whose larger body of work was cited in numerous places in the initial dataset. This approach helped to offset the difficulty of mapping the development of a given project over time, since particular projects evolved under various names and guises but often with some of the same principal researchers involved over time.

From the initial data gathering, I began to outline the relevant solution space and understand distinctions within design of collaboration support environments. Here, I selected systems and approaches to analyze in more depth by narrowing in scope, as the breadth of the problem definition at the outset identified many collaboration frameworks which could be treated as relevant. At this juncture I selected explicit filtering criteria to limit my results, and sought to group apparently similar approaches to the problem space.

**Categorization and filtering.** In assessing the sources discovered in the initial review, I sought to apply filtering criteria to limit the breadth of the final analysis. Initially, when deciding what general types of approaches were most relevant to the project at hand, I used a somewhat vague criterion of “fit” for the problem and solution articulation relative to the particular problem and solution space that “Hmmm…” was defined to address. In particular, I looked first for a match between the problem articulation in the reviewed materials and that of “Hmmm…” given above. As it turned out, in some cases the discovered literature had elements in its articulation of a solution that appeared to match with the “Hmmm…” approach, despite not necessarily having
an equally clear match in the problem articulation. As I proceeded, I applied increasingly specific criteria to narrow down the solution space to a manageable subset of the relevant technologies. This was made possible as I discovered major distinctions that helped clarify the problem space.

There are various angles taken in the research literature to addressing the broader problem space of technological support for open-ended and reflexive communication and design. Below, I will discuss the distinctions that emerged in this study that helped to frame and delimit a particular space within which to consider the “Hmmm…” solution and its possible implementations. This space, as developed here, is one of many ways to approach implementing the proposed functionality, however it represents a site where feasibility can at least be reasonably considered at a broad level.

Among the possible levels at which to analyze technologies, the level of user-intelligible affordances was the chosen emphasis here, and systems that address this level within published documentation were selected for consideration. Another way to frame this is to describe user-intelligible affordances as that set of entities or constructs that could be defined within documentation either ahead of time or by users after experiencing the system. This is the level of intelligibility at which the “Hmmm…” system was defined in the previous chapter, and it is the level at which the inquiry is conducted as well. In the course of undertaking this review I discovered ways that different layers of a system are entangled, as technical decisions from the lower levels of implementation play a central role in the way that users experience the affordances of a system. As defined above, the level selected here was the user-intelligible interaction architecture with the acknowledgment that its underlying code, resource environment,
and hardware implementation are intertwined to various degrees in considerations of open-endedness, extensibility, support for multiplicity, support for redesign, and other concerns.

Reviewing the data generated from the search outlined above, I filtered the readings into groups and narrowed down to a set of specific systems under consideration based on the general criteria above. I then performed an initial round of deeper readings, developed category distinctions within the solution space, and rated candidate systems along the dimensions of resonance with the proposed approach, specifically seeking to draw out from each technology the following: (a) general articulation of a problem space and a solution approach, (b) systemic affordances for interaction, (c) systemic affordances for redesign, (d) apparent limitations relative to “Hmmm…”, (e) apparent benefits or take-away to “Hmmm….”, and (f) unanswered questions. I found that the latter three were difficult to discern clearly from the initial reading in many cases.

It took this additional analysis of candidates to get to the depth that allowed me to narrow down to only a few specific approaches that had a strong relevance to the larger problem domain. Applying the aforementioned analysis to systems that passed initial filtering, I used the results to perform another act of narrowing down that left only those technologies that were ranked highest or “most in common with proposed approach” in terms of their problem articulation, comprehensiveness of approach, richness of affordances for interaction and redesign, and benefits or takeaway, and that were ranked least problematic in their limitations or unanswered questions.

Because there were multiple dimensions upon which to rank systems, and because systems and the research associated with them often had an evolutionary life that contained
multiple iterations and different foci, I did not attempt to arrive at a single reduced score for each system. Instead, I ended up selecting the systems that had the strongest positive characteristics across critical categories, and in some cases settled on one particular system or framework as an exemplar of similar approaches. While I realize this was not the most empirically rigorous way to derive a measure of relevance and determine a relevant data set, this approach was rigorous from a designerly perspective, in seeking to pragmatically determine a set of relevant artifacts that address a similar problem space and overlay these artifacts onto the problem space articulation that directs the design process undertaken in this thesis.

At this stage, I had identified seven candidate technologies that appeared to possess unique approaches to, and a high degree of resonance with, the problem space and proposed general solution articulated in the previous chapter. In selecting these, I used all the criteria above, however I placed an emphasis on similarity of problem articulation, deep affordances for redesign, and the appearance of tangible benefits or takeaway with regard to the “Hmmm…” system, considering good matches in these areas more valuable than other criteria. These technologies did not necessarily operate at the same level in addressing the problem domain, and I chose this set partially to include some diversity of approaches in order to draw a richer picture of the solution space.

During this phase I also reviewed further literature about each candidate system and attempted to parse out the underlying computational architecture and emphases of each. Considering these systems side-by-side, two approaches emerged as having more comprehensive relevance to the problem space, and in the end I chose to analyze these two systems in more depth, at the expense of a broad analysis of a bunch of systems. This was done in order to limit the scope of the inquiry, but also due to the difficulties in performing a direct comparison
between systems, since, as mentioned above, various approaches address the problem space at various levels and could be potentially complementary. Also, as mentioned above, lacking the knowledge about the underlying technical implementations of each system and their potential compatibility and technical feasibility, it was difficult to relate different approaches to one another directly, to see if and how insights from them could be combined together. This was an unfortunate limitation that really made it difficult for this type of technology analysis to make more direct claims about the feasibility of the “Hmmm…” approach.

I analyze these two systems with regard to the research questions asked earlier, namely, whether these approaches provide affordances for a constitutive metamodel system such as that proposed via “Hmmm…” . These general affordances that the “Hmmm…” system is based upon form the basis for identifying and analyzing relevant affordances of the systems under consideration. The particular reifications at the user-intelligible level for any given approach provide the point of departure for analysis. In particular, I looked to identify the structures that a system reifies within its model of entities, environments, their relationships, and the interactions and transformations among them. How are these reifications implemented, and what affordances are provided by virtue of this architecture?

Below, I will briefly discuss some major categories that emerged in this review to organize the larger solution space. Next, I will briefly consider the five systems that were considered as candidates but not chosen for final analysis. I will then introduce and analyze the two systems selected. I will conclude the chapter by discussing the ways that this review and analysis may inform the design of a computational “Hmmm…” prototype.
**Major relevant areas.** In this section, I will briefly consider the major categories of technology that were discovered within the set of relevant technologies identified in this review. First I will describe potentially relevant categories that were filtered out of the final review. Next, I will describe categories that were included and briefly discuss the relevance of each to the problem domain.

The review initially centered on contemporary literature from the last five or six years. As can be seen clearly with the emergence third wave of HCI, a focus on particular situations has led to more context-specific support for interaction. This includes a move away from foundations of interaction, focusing more on specific domains and contexts, such as particular situated practices, as well as a general move to embodied and ubiquitous computing perspectives. Of course, the literature reviewed in general tends to value a practicality that calls for direct use-case applications for any given ideas and theories. Generally speaking, these are values that may also limit the breadth of domain to which any given solutions are applied.

Because “Hmmm…” is an infrastructure-centered design, I selected systems based on their breadth of coverage relative to the problematic outlined above. As a result of this, many of the frameworks that have some relevance to the problem space articulated above are limited by their domain specificity, supporting collaboration and design work only within a given (and limited) problem space corresponding to some set of objects, processes, and constructs relative to a particular domain. I eliminated a variety of candidates at an intermediate stage due to domain specificity because I felt that they would unnecessarily limit the possibilities of a communication and design framework, though I did not directly assess the underlying schemas and structures. This included various systems that were clearly addressing an overlapping problem space, however limited themselves to a particular domain of practice. This might be seen as a category
in itself relative to the current inquiry, something like “domain-specific collaboration support environments”.

The literature on end-user programming, or EUP, that I discovered generally was filtered out of the final review for similar reasons of domain specificity (e.g. Ahmadi, Jazayeri, Lelli, & Repenning, 2009; Warth, Yamamiya, Ohshima, & Wallace, 2008). There are obvious connections between end-user and participatory design practice and the problem space outlined in this thesis. However, end-user development technologies discovered in this review generally did not extend affordances for redesign to the structure of the system itself or the component architecture to quite the same degree seen in other categories of systems such as virtual environments, groupware toolkits, and some middleware approaches.

Because of the criterion that technological solutions afford deep redesign that transcends domain-specificity, my focus moved toward open-ended infrastructures that support methods of computational reflection. Reflection is defined as “the activity performed by a computational system when doing computation about (and by that possibly affecting) its own computation” (Maes, 1987, p. 147). Prominent examples are found in the work of Gregor Kiczales, who was one of the principle designers of Open Implementation (e.g. Kiczales, 1996), the Meta-Object Protocol (e.g. Kiczales, Des Rivières, & Bobrow, 1991), and Aspect-Oriented Programming (e.g. Kiczales et al., 1997).

Generally speaking, the value of computational reflection is in the flexibility and extensibility they can provide for evolving systems (Dourish, 1995). Dourish has also implemented reflection in the context of CSCW toolkits via Propsero, utilizing Open Implementation and proposing other reflection techniques to address the needs of infrastructure
flexibility in the context of fluid use (Dourish, 1998). From middleware perspectives, computational reflection, is argued to be a useful approach to manage components and interactions between them in terms of evolution and re-use (Tramontana, 2000) and distributed reconfiguration of open, adaptive middleware (Grace, Coulson, Porter, & Blair, 2006). I initially researched computational reflection as a category within this analysis, however most of its concerns are beyond the limited scope of this inquiry, and are omitted here due to space limitations.

Three somewhat overlapping categories emerged as encompassing most of the systems that passed earlier stages of filtering. These categories are (a) CSCW or groupware toolkits, (b) collaborative virtual environments or CVEs, and (c) component-based middleware. These categories are rough distinctions; as the research progressed, I discovered that toolkits and CVE frameworks might themselves be categorized together under the heading of middleware. Taken together, a distinct focus emerges on supporting situated activity in the context of evolving resources and needs. I will discuss each of these categories briefly below.

A groupware toolkit is an integrated set of development tools and reusable modules that support developers in constructing situation-specific groupware applications. According to Dourish, “the goal of any toolkit design is to provide generic, reusable components that are applicable to a wide range of applications (1998, p. 111). While there are various ways of thinking about toolkits, Greenberg and Roseman (1999) define four components that any toolkit should include: these are (a) a runtime architecture, (b) some groupware programming abstractions, (c) groupware widgets providing specific types of functionality, and (d) session managers. CSCW toolkits can themselves be seen as collaboration support environments made up of tools for constituting virtual interaction, and the design and testing literature on CSCW
toolkits considers relevant problem spaces of flexibility, extensibility, and redesign. This literature raises important questions about decisions made in implementation and how they can affect the subsequent affordances of groupware toolkit systems, particularly around the tension between flexibility and reusability (Dourish, 1998; Dourish & Edwards, 2000; Greenberg & Roseman, 1999).

Generally speaking, CSCW toolkits have an intended audience of developers rather than end-users, however, and this is one limitation of this approach relative to the problem domain articulated above. CVEs, on the other hand, are largely end-user environments, although the frameworks that support them may consist of developer constructs not intended for direct end-user consumption. The category of CVEs includes text-based and 2D environments along with 3D simulations of physical space, sharing the characteristic of emulating certain elements of the physical world to provide an intuitive metaphor for users to interact around (Tomek, Gong, Shakshuki, & Giles, 2006). CVEs address the domain of situated interaction among multiple participants with new objects as possible design outcomes, and use some spatial metaphor to organize the collaboration of participants. Some systems represent large-scale attempts at defining the infrastructure for a CVE, while others opt for a “microkernel” approach that leaves most of the system’s heavy lifting to pluggable components (e.g. Hubbold et al., 1999; Oliveira, Crowcroft, & Slater, 2001). The 3D virtual environment was a major area of research in the late 1990s and early 2000s, and it was this sort of CVE that was discovered and considered in this study.

Another relevant category of technology is middleware that brings together existing systems, applications, devices, and other forms in order to interoperate. According to Issarny, Caporuscio, and Georgantas, the primary function of middleware is “to overcome the
heterogeneity of the distributed infrastructure” (2007, p.1). Generally, middleware focuses on coordination and meaningful interaction among components which are heterogeneous to each other, although this is usually interpreted very practically to mean difference at the level of device type, platform, and operational semantics. Questions raised by the variety of relevant literature falling under the general category of middleware include ways to address interoperability of platforms, protocols, and component devices or applications within collaborative CVEs and toolkits (e.g. Ciampi, Gallo, Coronato, & De Pietro, 2010; Wang, Dorohonceanu, & Marsic, 1999). Middleware approaches are oriented to supporting the coordination and interoperation needs of an ecology of components forming an interactive system.

The major limitation of applying these approaches to the problem domain outlined in this thesis is the lack of conceptual flexibility built into conceptions of components in the middleware paradigm. That is, most literature on middleware generally reifies the domain to which it solves the problem of interoperability to include particular classes of physical devices or software applications, and thus is not itself oriented to including within its scope more ambiguous and human encapsulations. The implications of this, along with the evolution of the researcher’s understanding of this category and its relation to “Hmmm…” will be discussed in more depth later in this chapter.

The emergence of these three categories corresponds to a move to focus the solution space upon architectures supporting collaboration rather than end-user applications. Technologies considered here often focused their direct efforts on providing tools for developers. Some CVE approaches bring architectural considerations closer to users, however the underlying
architecture itself that could support affordances the “Hmmm…” system would require became the central focus of the inquiry.

**Candidate systems.** The seven systems that were considered in depth include the following:

1. The DARE meta-groupware system and its evolution into CooLDA and implementation in an Eclipse-based IDE known as CooLDev (e.g. Bourguin & Derycke, 2000; Bourguin, Derycke, & Tarby, 2001; Lewandowski & Bourguin, 2007, 2008).

2. The Deva family of VE systems (e.g. Daubrenet & Pettifer, 2003; Pettifer, Cook, Marsh, & West, 2000; Pettifer & Marsh, 2001),

3. The MASSIVE family of systems and related work by Greenhalgh, with a focus on MASSIVE-3 (e.g. Greenhalgh, Pubrick, & Snowdon, 2000; Greenhalgh et al., 2000; Greenhalgh, Flintham, Pubrick, & Benford, 2002; Pubrick & Greenhalgh, 2003).

4. The Croquet framework and related work in Cobalt and TeaTime coordination mechanism (e.g. McCahill & Lombardi, 2004; Reed, 2006; Smith, Kay, Raab, & Reed, 2003; Smith, Raab, Reed, & Kay, 2004)

5. The Fiia framework and Fiia.net application (e.g. Wolfe, Graham, Philips, & Roy, 2009; Wolfe, Smith, Philips, & Graham, 2010),

6. The Orbit collaboration system, and its predecessor systems wOrlds and ConversatinBuilder (Bogia & Kaplan, 1995; Fitzpatrick, 1998; Fitzpatrick, Kaplan, & Mansfield, 1996; Fitzpatrick, Tolone & Kaplan, 1995; Kaplan, Fitzpatrick, Mansfield, & Tolone, 1997; Mansfield et al., 1999; Tolone, Kaplan, & Fitzpatrick, 1995)
7. The Intermezzo toolkit defined by (Edwards, 1995; Dourish & Edwards, 2000; Edwards, 2005)

In the end, I selected Orbit and Intermezzo to examine in depth as generally comprehensive approaches to addressing the problem of open-ended, situated action, as these provided the clearest cases of affordances relevant to the proposed “Hmmm…” system. Among all approaches considered, these two seemed to provide the best combination of affordances for complex and layered communication environments coupled with deep affordances for redesign. Before discussing these directly, I will briefly describe the other systems under consideration so as to situate some of the insights gained in the subsequent analysis of Intermezzo and Orbit, which will also be introduced in this section. Next, I will analyze Orbit and Intermezzo approaches in terms of their affordances and constraints for a constitutive metamodel. Finally, I will return to a direct consideration of whether and how these findings have theoretical and practical implications for the proposed “Hmmm…” system.

**DARE.** The DARE metagroupware architecture evolved into a platform known as CooLDA, from which was implemented a collaborative development environment called CooLDev. The DARE model aims to provide an open-ended, activity-driven collaboration framework. DARE is founded very explicitly on the ideas of Activity Theory, and frames itself within research on Evolving Interactive Systems, or EIS. In particular, DARE attempts to provide a “favorable milieu for the evolutions management of different EIS while adopting a psycho-social-historical perspective founded on the Activity Theory” (Bourguin et al., 2001, p. 303).
DARE provides tailorability through the component approach, “glued” together by the construct of an activity and the infrastructure that supports it (Bourguin & Derycke, 2000, p. 278). Their approach allows users to link integrated components with particular roles within structured activities, and thus flexibly determine how tools will be used in practice. Furthermore, the metaarchitecture is implemented as “applets” that can be situated as components in parallel ways, thus allowing for redesign of metalevel mediation rules within “real co-operative meta-activity” (Bourguin & Derycke, 2000, p. 279).

Deva. Deva is one among numerous systems that attempts to address the needs of large-scale virtual environments by developing a microkernel that accommodates and moderates the functionality of pluggable components. In particular Deva is implemented via a client server architecture that applies a behavioral topology to solving the problems of application-specific representation, distribution, and computational execution. One intent of the Deva approach is to separate the computational semantics from the appearance or view of the environment (Pettifer et al., 2000).

Deva is based on entities which are coarse-grained programming objects that can communicate with one another. Entities may represent objects, properties of the environment itself, or abstract programming objects with or without direct representation to inhabitants of the environment (Pettifer et al., 2000). Deva makes the distinction between four sources of behavior relative to a given object, including multiple sources of inherent and environment-imbued behaviors, treating these as different components within a single entity. Entities are further divided into a single object and numerous subjects, where subject to object and object to subject communication form the basis for the material behavior of the entity.
**MASSIVE.** MASSIVE-3 is framework for CVEs that aims to support hierarchically structured objects, data consistency, heterogenous components, flexible interest management, and persistent worlds beyond a given creating application (Greenhalgh et al., 2000). Massive is explicitly designed in the 3D virtual world paradigm, yet attempts to go beyond a purely spatial model, drawing on the software platform EQUIP (Greenhalgh, 2002) designed for managing dataspaces in augmented reality that overlay digital and virtual data on a physical world.

MASSIVE-3 is based on a construct called locales, implemented as independent, local-only coordinate systems via distributed databases called Environments. The behavior of locales is governed in part by aspects, which are layered units within objects that govern the behavior of objects within a locale, and provide the ability to structure locales according to properties of their objects. Aspects are used to define the scope of behavior, ownership and membership, degree of compression and something they call “cost”, of objects, the latter of which could resemble salience and solidity as a strategy for interest management of resources.

**Croquet.** On the surface, the problem and solution articulation of the Croquet framework may be the closest parallel to the “Hmmm…” system, taking a very ambitious approach to reconsidering computational foundations of computing as situated within interaction. According to Smith et al. (2003, p. 3), the paradigm of computing that underlies current operating systems is based on a “closed individually focused system”. They argue that human communication has emerged as central to the experience of computing, yet communication “is still done through the narrowest of pipes” (p. 2). Their general approach is to build an architecture “from the ground

---

18 The construct of locales used here is distinct from that underlying the Orbit system considered later. With regard to MASSIVE systems, the specification of locales is that developed for the SPLINE system (Barrus, Waters, & Anderson, 1996).
up” that is focused on “deep collaboration”, and using the unique affordances of the Squeak language to compose the environment as a platform for both delivery and development with no essential distinction between the two.

Croquet is implemented in Squeak (e.g. Ingalls, Kaehler, Maloney, Wallace, & Kay, 1997), a derivative of the Smalltalk language that is capable of implementing transformations from within a running environment without recompiling (Smith et al., 2003). Beyond this, however, Croquet is defined as a straightforward 3D spatial environment, where spaces are the medium within which users can co-create and instantiate various implicitly collaborative objects. Croquet’s coordination architecture TeaTime (Smith et al., 2003; Reed, 2005), is based upon “replicated versioned objects coordinated by a universal timebase embedded in the communications protocol” (Smith et al., 2003). TeaTime appears to add a layer of multiuser message coordination and synchronization to the basic reflective message-object ontology of Squeak. This approach seems to provide a deep flexibility for the synchronization and coordination of behaviors that, by virtue of the underlying computational structure, can have direct reflective effects on the affordances of the situation. However, it appears that significant development of human-centered affordances for constitutive interaction have not been subsequently taken up within this stream of research, and the focus of its descendant Cobalt has been squarely in the spatial virtual world simulation paradigm. This was the main reason why I chose not to analyze this approach in more depth.

**Fiia.** The Fiia architecture and Fiia.net technology takes a distinct approach in that it focuses on defining an architecture of collaboration elements which allows end-users to literally build their own applications using a simple diagramming language to specify a collaborative situation, which can then be compiled via the Fiia.net toolkit to functionally constitute a
collaboration environment (Wolfe et al., 2009). Billed as a user-centered, application level, runtime-adaptable groupware notation language, Fiia affords collaboration design via annotations that constrain runtime choices of the system, configuring and reflecting how a system is actually deployed. Wolfe et al. argue that middleware traditionally does not provide the type of support needed for building groupware, particularly support for runtime adaptation with high-level conceptual language.

Fiia’s basic annotation language specifies various types of components with different functional behavior: stores, reactors, actors, and adapters. Mapped to construct actual systems via the Fiia.net toolkit, this annotation language allows users to create various affordances and constraints within what the authors call a “setting”, as well as connecting external systems into the infrastructure and providing for the synchronization and sharing of settings and their components. This approach allows for high-level design, where designers can build a scenario of proposed use through iterative snapshots and situation representations, to be transformed into code. Adding and connecting components to a scenario is supported in simple high-level ways, allowing fast changes even if they involve complex implementations. The Fiia approach was perhaps the most user-friendly of the subset of approaches considered here, however its representations of components appeared to limit the possibilities of expressivity within the designs to mechanistic dimensions of collaboration only.

**Orbit.** Orbit is a generic platform for collaboration support, prototyped over multiple iterations during the late 1990s (Mansfield et al., 1999). The Orbit project extends from two earlier projects, wOrlds (Bogia & Kaplan, 1995; Fitzpatrick et al., 1995; Kaplan et al., 1995; Tolone et al., 1995) and Converstion Builder (e.g. Bogia, Tolone, Kaplan, & Tribouille, 1993; Kaplan, Tolone, Bogia, & Bignoli, 1992). According to its designers, Orbit is eventually
intended as a “generic toolkit for constructing collaborative environments”, supporting both the
development of complete collaboration environments from scratch and the piecing together of
existing collaboration tools into an ad hoc environment (Mansfield et al., 1999, p. 368).
Mansfield et al. state that their eventual aim is for Orbit to operate as a general collaborative
desktop environment that is intended to become invisible or indistinguishable from the working
environment of everyday use (p. 374).

Orbit’s predecessor wOrlds system draws upon ConversationBuilder and its Obligation
system (Bogia et al., 1993; Kaplan et al., 1992), which contains a protocol specification language
loosely based on Speech Act theory, similar to the COORDINATOR system discussed briefly in
Chapter One (Kaplan et al., 1997). Conversation Builder aimed to support the contingent and
dynamic nature of cooperative work, focusing on specification of the formal dimensions of
collaboration, based on a fundamental unit of an action (Kaplan et al.). Drawing upon the work
of Anselm Strauss, the authors of wOrlds argue that such reifications are doomed to failure, since
actions change fluidly within real-world interactions (Kaplan et al.). They attempt to address this
limitation by introducing a cultural level alongside (and equal in importance to) the formal level,
drawing upon Strauss’ notion of social worlds as inspiration for the development of what they
call locales (Kaplan et al.). Going beyond a purely spatial representation, locales are “collections
of conditions which both enable and constrain action possibilities, that together with ‘action
in/with setting’, become context for further actions.” (Fitzpatrick et al., 1995).

Orbit draws upon wOrlds’ reflective architecture for the specification of locales, actions,
and processes. These elements are the constitutive features of the interaction environment, and
represent reifications of underlying code to coordinate the behavior of resources shared among
users, applications, interfaces. In particular, the Introspect language is provided as an interface
for the specification of each of these three core constructs. This will be discussed in more detail below.  

Orbit aims to expand beyond the wOrlds system in terms of affordances for multiple sites of concurrent activity, individual views and need for mutual awareness, and history or trajectory information (Mansfield et al., 1999). This is done by extending the notion of locales from wOrlds to include the wider scope of the locales framework (Fitzpatrick, 1998; Mansfield et al.). This is an attempt to relate a language for understanding sociality to a language for specifying collaborative activity and its support. Defined as a framework providing “shared abstractions” for understanding real-world interaction and for designing models of interaction support, the locales framework begins with the notion of a locale as combining the interactional needs of the social world with the “site and means” of collaboration (Fitzpatrick, p. 91). There are five elements of the locales framework: (a) locale foundations; (b) civic structure; (c) individual view; (d) interaction trajectory; (e) mutuality. These are meant to extend the notion of locales to consider further the relationship of locales to interaction generally (Fitzpatrick) and to develop the necessary affordances for situated contextual interaction in locales. Thus, the locales framework provides the broader meso and macro foundations for locales, as well as providing for and managing micro foundations of general coorientation, providing a structure of intelligibility to interaction. These will be discussed briefly below as well.

**Intermezzo.** Intermezzo is an architecture for collaboration developed by W. Keith Edwards. Intermezzo focuses on infrastructural solutions to the social and architectural

---

19 Many of the core technologies considered in the analysis of Orbit are significantly developed in the wOrlds project. For the sake of readability, I chose to describe the analyzed system as Orbit rather than some compound like Orbit/wOrlds which would better reflect the importance of wOrlds as the initial site where these ideas are developed.
problems of designing collaborative applications in the context of an evolving space of applications and objects. In particular, Intermezzo seeks to provide context-awareness that balances the differing semantic needs of humans (e.g. interpretive flexibility) and machines (e.g. interpretive rigidity). Edwards argues that what needs to be addressed with regards to design of collaboration support environments are coordination needs, which are largely based on social factors (Edwards, 1995). With regard to the wicked problem of coordination among users, applications and data entities, the affordance of a common language for coordination that is external to the individual applications represents an “environment-centered” solution, as opposed to application-centered (Edwards, pp. 18-19).

Edwards argues that an infrastructure for sharing what he terms coordination information is both absolutely necessary and may call for a different infrastructure than application information-sharing itself. Intermezzo creates a framework for coordination that allows applications to develop semantic interpretations and constructions of data which are application and situation-specific, yet attach these to an infrastructure so that they can be shared in future interaction around objects. This is done in a particular way so that applications drawing on the shared infrastructure do not need to know all aspects of possible context, yet can draw upon these and add their own context to persistent data as well.

Intermezzo’s conceptual architecture is based on the premise that the activity is a fundamental unit, and consists of components analogous to Smith’s (1996) unit of registration. Intermezzo’s environments are implemented as distributed object systems that are schema-less aside from the basic components of an activity, and that provide specific affordances for developing application-specific schemas. Such environments have their own internal relations structuring complex contexts for interaction. The Intermezzo framework extends basic tuple
space models, providing a unique set of strategies that allow environments of interaction to embody various important dimensions of human intelligibility: ambiguity, evolution, complex notions of sameness, and extensibility. These will be discussed below.

**Findings and Analysis**

Here, I will enter into a consideration of the particular affordances of each approach as they are applicable to modeling a constitutive relationship between communication and the design of a situation and its content. These considerations will fall broadly into the affordances for interaction and affordances for redesign of the system. Finally, I will summarize the findings in terms of what they can teach about the feasibility of developing collaboration support environments based on the general goals and specific features proposed via “Hmmm…”.

**Interaction affordances in Orbit.** Much of Orbit’s underlying conceptual architecture draws from the model of locale, action, and process specification first developed within wOrlds and extended for Orbit via the locales framework. In this model, locales are the site where interaction is structured and occurs. Entities that can interact within and across locales include users, external systems or data objects, and internal encapsulations of locales, activities, and processes, each with their own attributes from which they derive actionable elements of the architecture. The environment is constituted in reifications of locales themselves, as well as activity, users, processes, and what the authors call “particulars” (Tolone et al., 1995). All of these are actionable components encapsulated as objects, which can be structured with behaviors and attributes as described below.

The locales framework attempts to overlay a rich and complex framework defining situated social action via five aspects: locale foundations, civic structure, interaction trajectory,
individual views, and mutuality. Each of these aspects represents a specific set of affordances, together constituting a highly-programmable localized interaction situation. Theoretically, various properties of these aspects could be attached to constructs representing elements of a situation, with properties defined along multiple dimensions. Some of the proposed properties correspond directly and indirectly to particular meta-aspects proposed in “Hmmm…”. For example, the category of interaction trajectory proposes to represent temporal relationships and dependencies, as well as a past, present and future attributes, within each element containing such an interaction trajectory, which is to include not only locales but people, particulars, and events as well (Fitzpatrick, 1998).

Taken together, the properties and dimensions populating aspects of the locales framework are proposed as ways to make sense of and program relationships among elements in order to constitute a rich activity space. The framework is primarily situated as a tool for thinking about the broad requirements of a culturally and well as formally tailored group workspace. The properties and dimensions found in Fitzpatrick’s (1998) work are not described explicitly to operate as data constructs, however, so I will not consider them further here, opting instead to focus on the underlying locale architecture as largely specified in the wOrlds system.

In Orbit, processes and actions work together with locales to form the basis of coordination. Each of these three constructs is specifiable via the Introspect specification language (Tolone et al., 1995). Within the framework, each of these core elements are reified constructs with particular properties defining their identity, behavior, and relationship to other entities. Actions, for example, are user-intelligible, potentially locale-specific encapsulations of a sequence of smaller actions or basic steps of code execution. The action construct is populated by properties such as description, parameters, action model, bindings list, and code, functioning
as “a complex wrapper around an action sequence” (Tolone et al., p. 62). These properties endow actions with their own identities, and create the potential for environmental conditions to trigger, or prevent, actions from executing under various conditions.

The specification of locales, actions, and processes constitute much of the flexibility of the collaboration ontology in Orbit. Locales contain attributes corresponding to membership in the locale, as well as domain-specific roles, actions, and processes. Locale attributes also contain code that instantiates locale behavior, such as specific action sequences for initiation, termination, adding participants, etc. Process models are represented as state charts, containing events which represent actions in a locale (Tolone et al., 1995). Actions, when embedded in a process model, are executed according to a multiphase structure whereby deactivation and activation sequences are triggered from within the action, allowing each of the phases to potentially transform locales and process models reflectively (Tolone et al.).

Using the Introspect specification language, actions, processes, and locales are all programmable objects within Orbit. Modifications made within a situation to actions, processes and locales, can come to be inherited by current or future elements of the same type. Users are asked to specify the scope of changes they make within the specification of entities, in terms of whether changes should extend upward to meta-objects defining the object type, and thus outward to other specifications of the type. This will be considered in the next section addressing the redesign affordances of Orbit.

Redesign affordances in Orbit. The Orbit system also opts for a highly-developed architectural solution for reflection. Orbit is implemented in a variant of Smalltalk, a programming language developed as an exemplar of the possibilities of object-oriented
programming (Kay, 1993). Modern variants of this language support reflection and redesign occurring at runtime, at the language level. Theoretically speaking at least, this allows changes to the runtime environment of Orbit to be created and computed from within an existing interaction situation, in a similar way to that put forth in Croquet using the Smalltalk variant Squeak. 20

Starting with the wOrlds project, a specification language is utilized called Introspect that uses a meta-level architecture to enable “radical tailorability” (Tolone et al., 1995). Their approach utilized computational reflection, with Tolone et al. citing Maes (1987) for their definition. In Introspect, a three-level meta-architecture is proposed: meta-specification, specification, and instantiation. The meta-architecture is applied to locales, actions, and processes. Each top-level meta-specification can be meta-object to multiple second-level specifications, each of which is in turn meta-object to a single instantiation. This model thus provides what Matsuoka, Watanabe and Yonezawa (1991) call a “hybrid group reflective architecture” approach combining individual-based (one object to one meta-object) and group-based (many objects to one meta-object) relations (p. 234). What differentiates this architecture from a normal inheritance (specification-prototype) relationship, according to the authors, is that changes made to instantiations and specifications can extend upward to meta-specifications. Reflection occurs in the specification (and meta-specification) of locales, which can dictate behavior of the system itself; for example, locales that are tailored to using Introspect (Tolone et al.).

20 I did not verify whether Orbit provides this underlying capability, or whether, like Croquet its programming environment and use environment completely overlap as is the intention of Croquet.
Summary and brief analysis of Orbit. The specification of locales, actions and processes working together creates a context-dependence to actions, in line with a constraining and affording relationship of environment to action. In Orbit, activities begin as a particular instantiation of a type, however as they are modified, their relationship to this type can be defined and redefined within interaction. This allows context-specific and broader effects to both be part of the infrastructure. By constraining activities in terms of their locales while simultaneously providing the ability to transform locales and contexts via actions, Orbit meets one basic requirement of a metamodel system as outlined above.

The ontology of locales, actions, and processes still requires the addition of a layer of coordination applications to make use of its ontology. The architecture added by the locales framework attempts to deal with the richness of this coordination, and promises a much richer set of social affordances that could be encapsulated into properties of the reified structures contained in this model. However, a strong implementation of the locales framework within an Orbit prototype, to my knowledge, was never developed (Mansfield et al., 1999; Dourish, 2005). Issues with implementing sufficiently-flexible aspects of the locales framework remain unresolved in Orbit prototypes, with support for interaction trajectory cited as a particularly significant challenge (Mansfield et al.). It appears that development of Orbit sought to address these issues in the future, however development stopped short of reaching feasible solutions.

Interaction affordances in Intermezzo. Intermezzo’s built-in structure involves specifying activities as compound elements that contain as components, other entities. This model eschews location-based identity in favor an activity-based construct analogous to Smith’s (1996) model of registration: subject, verb, and object elements are connected to any articulation of an activity. These are analogous to the user, the application (or process being performed), and
the artifact being operated upon. Within Intermezzo, various aspects of coordination, including awareness, session management, and access control are implemented directly by drawing upon this representation of activity.

Each of the activity components are resources with attributes that provide information about them that can be used by other agents. These attributes enable important coordination functions by providing both identifying and relational information about the resource. For example, the awareness about entities is based not on a spatial or place-based proximity, but rather on pattern-based discovery and notification around activities, as the components of activity are published to the shared dataspace. Edwards (1995) likens this on a simple level to a “behaviorist” model of awareness, where we only know others by virtue of their actions (p. 50).

Edwards similarly implements a light-weight approach to session management that is also derived from the basic construct of activity (Edwards, 1995, p. 69). In general, Intermezzo moves away from explicit session management, arguing instead that the activity of rendezvous and interconnection are inherent outcomes of interaction and need not be formally entered into. In this way, Intermezzo treats interaction episodes as a given feature of activities rather than as special-case containers around the activity, which may perhaps be in clear contrast to the approach of Orbit, which could be characterized here in Edwards terms as an implementation of locality-based implicit session-management. In this approach, the virtual co-location of users provides a “built-in” mechanism for rendezvous (Edwards, p. 71).

Edwards approach is to use activity instead as a foundation for implicit session management. In particular, the broadcast attributes of activity allow a session management service to detect and adapt to potential situations of rendezvous and collaboration. The potential
for collaboration is tied to “overlaps or confluences in the total set of activity information
published by all applications across the network” (Edwards, 1995, p. 73). Edwards argues that
infrastructure for implicit session management must support the provision of new functionality
upon detection of collaboration possibilities, and that such functionality must be arbitrarily
extensible in ways unplanned by the initial designers.

Intermezzo seeks also to implement a flexible policy framework to govern fluid access
table to resources. In particular, Intermezzo defines policy via roles, which are dynamic and
can change based on the state of a given user’s world. Access control in Intermezzo is done by
roles that are specified using predicates based on the context of the situation. This is in contrast
to using an explicit membership list. This allows contextual roles to be dynamically created and
assigned. Crucially, the features of awareness, session management, and dynamic roles operate
as a “closed loop in the sense that all three coordination features not only build upon each other
but also feed each other”, making possible a “synergistic effect” (Edwards, 1995, p. 88).

Edwards also implements a tuple space distributed shared memory, encoding activity-
based components and attributes as minimal, coarse-grained infrastructure within this model to
provide a simple way to organize and relate subjects, objects, and actions. However, the major
focus of Intermezzo is supporting the ability to create new structure to the relations between
entities without ‘breaking’ the infrastructure (i.e. necessitating its underlying redesign). In order
to support this, Edwards provides mechanisms for complex structuring of resources via the
addition of properties and embedding of functionality within these entities.

In tuple spaces, each tuple can be thought of as a slot for data. Intermezzo extends the
basic tuple space model in unique ways, namely, by implementing extensible, multivalue, and
scoped slot architectures to supplement a general distributed database tuple space model towards supporting extensibility, ambiguity, and complex notions of equality. The ability to add new layers to the infrastructure is combined with mechanisms that support canonicalization of resources, to accommodate multidimensional identity in persistent resources based on various activities’ relationships to a shared material referent. I will discuss each of these below.

Extensibility is defined by Dourish and Edwards (2000) as the ability to add new components at the same level of existing primitives. To support extensibility, Intermezzo uses a schema-less extensible slot model, where no type model controls slots and thus different applications can add slots without interfering with existing slots. Thus, applications can add their own schemas on top of referents in ways that can be re-usable to other applications that are configured to share the schema, without requiring any other applications to adhere to said schema, and without the infrastructure needing to understand the schema. The implication is not simply that two users can see different views when looking at the same referent, but that two entirely different semantics can be applied to the same referent, giving it two different functional specifications that do not contradict or cancel one another out.

Intermezzo also allows its objects to be tied to persistent multidimensional entities via a reference-checking and merging process called canonicalization. This ensures that certain patterns reference a single object that stands for a particular persistent real-world object. In particular, the approach is to allow applications to impose custom semantic constraints, so as to develop a rich layering constitutive of a single subject, application, or object involved in multiple activities. Canonicalizations act as centripetal forces pulling together different representations of a user into a shared referent of a subject (much like the subject-object as conceived above). They work by checking attempts by applications to create new resources against identity predicates,
which filter into an encapsulated resource all instances of new resources matching patterns that correspond to the persistent identity of said resource (Edwards, 2005).

To explicitly support contextual ambiguity, Intermezzo allows multivalued slots, allowing a single attribute to be associated not with a simple value but with a collection of values from different sources. This enables multiple different views, uses, effects, influences, or behavior to exist for any given shared referent. To support complex notions of ‘equality’ and sameness, Intermezzo allows slots to be scoped, creating different sets or levels of access to a given resource that can co-exist simultaneously. An example given by Edwards (2005) is the different scopes of interest that different applications (such as system manager, source code control, and editing tool) have upon a file being edited. The goal here is “dynamic multigranularity interpretations of data” (Edwards, p.460).

**Redesign affordances in Intermezzo.** Intermezzo does not cite constructs from computational reflection directly, though it also clearly provides deep redesign capabilities through its general architecture of objects. Intermezzo defines an interface to the runtime system and allows for code to run downloaded into the runtime, and for runtime events to trigger code in various ways. Applications in Intermezzo can download code into the runtime system that can be triggered by any change in the environment and can transform applications or the toolkit itself. This provides a powerful way to reflectively redesign, and appears similar to the paradigm of Aspect-Oriented Programming (Kiczales et al., 1997) in its general orientation, in that it allows programmers to address concerns that cross-cut particular logical modules within the system.
In particular Intermezzo provides notification services, a foundation-level API, 
“coordination-specific” APIs, and scripting as particular ways of interfacing with the system and 
runtime environment. Notification mechanisms involve specifying events, triggers, and 
embedded computation make runtime environment reflexive so that actions occur based on 
conditions. Events are based on key points, which cause the assessment of whether event 
conditions are met. This creates lots of flexibility, as events result from messages, triggers 
execute immediately and are tied to specific resources, making them potentially active at a level 
not mediated by overarching ontology – this allows both pattern-matching and direct versions of 
execution. Embedded computation allows embedding of code that executes with a user’s 
permissions on the server side or on other clients, rather than in the client application of the 
creator. This approach generally allows for the separation of performance-critical code from 
flexible script code, thus enabling development of what Edwards calls “hybrid applications” 
whose own code is supplemented by a scripting language (1995, p. 103). Also, embedded 
computation provides execution ‘on behalf’ of a user, serving as a proxy for an offline user, as 
well as a general mechanism for notification.

Summary and brief analysis of Intermezzo. In Intermezzo, a complex multiplicity of 
structural semantics share a partially overlapping dataspace. Agents can reconfigure the 
environment as well as additively modify it. Activities are the constitutive unit of user-
intelligible structure, based on a coarse-grain activity model that resembles Smith’s registration 
construct. However, beyond this coarse-grained structure, the context in which an activity and 
its component registration reside are not structured a priori in terms of a place and a set of 
affordances. Rather, the intent is for the infrastructure to support multiple emergent
configurations of such contextual situationality, driven largely by the applications interacting within the system.

**Discussion of Results**

Each system has particular ways of modeling interaction, and particular levels at which communication and design work can be constitutive within their model. Drawing upon the analysis above, we can see some core distinction between the two systems. These center around the ways that elements of interaction are organized, particularly in terms of the schemas overlaid upon elements of a distributed object system, activities that affect them, and the entities involved.

**Configuration as constitutive.** If we first ask about what ways these systems are constitutive models of interaction, a common metaphor can be found underlying both approaches, as well as many of the other candidate systems. In their own way, each treats configuration (of users, tools, and resources, in a locale; of activities and components) as the basic form of articulation that is constitutive of surroundings. Thus, *configuration-as-constitutive* could be seen as a central metaphor in collaboration support environments and middleware, as they aim to support ad-hoc configurations of elements being specified in order to constitute specific contexts of interaction. The difference is in how configuration is organized via an underlying ontology describing the fundamentals of configuration.

**Properties as relational boundaries.** A second similarity between these two approaches is their use of properties or attributes of elements to define the functional behavior of elements and their relationships. In both systems, the fundamental elements are structured via particular properties or fields that define them, which can be specified, and in the case of Intermezzo, can
be extensibly added to elements, providing for new ways of relating elements together. Similar in some ways to the notion of meta layers proposed in the previous chapter, properties are used to provide identities to elements and manage their effective behavior within the space of interaction. Again, the distinction between the two systems is largely in how these properties are organized in the system, and what affordances and constraints each respective properties-based organization scheme provides.

**Organizing configurations: Locales vs. activities.** Orbit offers a conceptual model to specify and organize (and reflect upon) shared resources and to specify and organize activities and processes occurring at the level of real-time situated collaboration among humans. In Orbit, the general approach is to organize coordination around the construct of the locale, providing structured building blocks for the elements of action and process. Locales, activities, and processes are created, invoked and evolve as specifications of organized interaction, with preset properties that define the functionality of these elements. Locales are a very rich concept within the Orbit framework, and can be applied in numerous situations, however the question emerges as to whether locales are the best way to understand all configurations among elements.

Orbit provides individual views of a single predefined context container known as a locale. Locales, and the ontology they presuppose, may not be the most useful representation system for all types of configurations that could be thought of as collaborative and communicative. A certain notion of culture which encapsulates based on symbolic relational practices is presumed, without an ability to really involve multiple layers of culture, since the multiplicity of realities and uncertainty implied in theorizations outlined above problematizes the notion of a single locale. In addition to the question of flexibility this container itself provides,
there is a question of the degree to which coordination features such as mutuality and individual views can reflexively affect the properties of the resources they affect.

Within Intermezzo, the coordination functions are integrated into the underlying coarse-grained ontology of activity, and much more of the specific configuration of relevant context is left up to the participants. Activities contain subject, verb, and action components, but do not specify a place. Thus, Intermezzo represents an environment-centric approach to collaboration that operationalizes its environment in a different way than Orbit. In particular, Intermezzo’s solution to the problem of flexibility is to focus on the activities that are occurring, allow applications to evolve the structure of place as they “impose semantic constraints on the data store” (Edwards, 2005, p. 449). It might be said that Intermezzo offers an activity-based conceptual model to organize the coordination of applications, which are themselves largely responsible for the modeling and representation of shared resources and processes. In other words, it is up to applications to work together with the support of the infrastructure to create anything placed as the container for a given interaction. The actual user-intelligible ontology is less determined at the level of implementation, and in terms of what is set out as the basic model of activities, their components, and the active role of properties in determining the behavior of entities.

Organizing configurations: Placed vs. placeless. Perhaps, then, one significant distinction in the problem space can be thought of as one of place-based versus placeless organization of interaction at the level of middleware. This could also be conceived of as centralized versus decentralized ontology of place, defined as the scope of responsibility given to any given ontology of place within any given coordination infrastructure. In centralized models like Orbit, place is coupled to the middleware application/system and becomes a central
structural determinant. In decentralized models, developing models of place is the responsibility of the elements themselves, supported in their coordination, sharing, canonicalization, and reuse needs by the middleware, which they can also influence and transform.

The term placeless appears in the Placeless Documents system described in Dourish et al. (2000), which displays similarities to the Intermezzo approach. Intermezzo, like the Placeless model, uses the attributes or properties of entities as the primary interface for document management and collaboration. User-specific properties enable multiple separate encapsulations of situated behavior to coexist without breaking the infrastructure. Furthermore, properties can be active and control resource behavior, thus adding functionality to the system via embedding code within shared resources. Combining these together allows users to configure behavior of resources via a shared interface to properties that simultaneously supports individual interpretations (views, layers of relevant context, etc.). While the Placeless model is centered on documents and files, Edwards extends the idea of schema-less property-based management to the representations of resources, agents, and interaction processes, with a framework of minimal complexity intended to support emergence of collaboration ecologies.

In Orbit, place defines the interaction ontology, and elements of coordination such as session management, mutual awareness, and access policy are built around this notion of place. As a result, all coordination mechanisms are required to be built to fit with the existing ontology. Issues with this approach can be seen in the limitations cited for Orbit, which correspond directly to some of the core areas Intermezzo addresses. In particular, Mansfield et al. (1999) describe existing prototypes as being limited in their flexibility, citing specifically the need for a flexible access and permissions model and better awareness support. As described above, Intermezzo
has integrated these concerns flexibly by connecting awareness, access, and sessions directly to the properties of activities and their components.

Furthermore, Orbit prototypes realized only a limited notion of the locales framework, with somewhat inflexible implementations of the locale construct. According to Mansfield et al. (1999), locales are singularly delimited, closed spaces in Orbit prototypes, that is, objects within a locale are accessible to the locales members only. Courtyards between locales are the only way to implement boundary objects. In Intermezzo, we get closer to a model where everything is in a sense a boundary object, and the place of interaction is defined more by the ways that boundary objects are invoked than by the ontological particulars of the location of the object.

**Organizing configurations: Centralized versus decentralized ontology.** The move away from strict ontology imposed on elements to general associative frameworks represents something more in line with a constitutive view of how the world is organized. A major goal of Edwards work with Intermezzo, as well as other work on *recombinant computing* (e.g. Edwards, Newman, Sedivy, Smith, & Izadi, 2002), is to allow flexible, ad hoc interoperability among applications and components with only limited a priori knowledge of one another. In systems like Orbit, a more detailed ontological framework is proposed as a basis for relationships, with overlap and interdependency between among its logical parts within implementation. However, Mansfield et al. (1999) note that the design trends through the evolution of Orbit have been away from a “completist, design-centered system and towards a piecemeal system which is open and tailorable enough to be adapted to the context in which it is placed” (p. 374).

While Orbit provides a lot of useful affordances for the sorts of dynamic ontology proposed, its use of a locales model may overdimensionalize interaction to a specific language of
context. While the meta-architecture of Orbit provides for reflection, it may be that this model is limiting insofar as it provides a specific three level infrastructure that, in practice, appears to be fairly rigid. While the Introspect capabilities make possible a lot of expansion within this model, the overall approach also presumes to be a general ontology governing all interaction, and thus the underlying notions of locale, action, and process (as well as reifications of users and components) are less capable of being fundamentally transformed or reinterpreted in terms of their meaning; the metaphorical extensibility is limited in some sense by the ontology.

Edwards approach is a somewhat less of an ontological solution to reflection than the three level infrastructure of Orbit, I would argue. By allowing applications to encode objects extensibly with new behaviors that do not necessarily cancel out prior ones, entirely heterogeneous applications could draw upon a shared infrastructure for ongoing redesign of collaboration functionality and content across heterogeneous contexts. Orbit, by virtue of its ontology, in some sense homogenizes the ways that redesign can happen, but also the ways that collaboration can be represented.

Intermezzo’s framework makes the innovative move of defining the infrastructure for awareness and visualization to share a common activity model with the infrastructure for session management, access control, redesign via code embedding, etc., and then to implement this model within an extensible properties-based semantics. Via their properties, activities are reflexively situated and play an active role in the coordination of their context, rather than simply being contained within a predefined context. Adding new semantics without breaking compatibility is a high-level infrastructural strategy to support development of vocabularies of situated activity that can have structuring effects on the runtime environment.
This is a less ontologizing approach to define interaction support and teaches us a different way to think about the modeling of situated action; namely, by letting the entities involved add their own semantic modeling to a common infrastructure that supports reuse of these semantics. Taken together, Intermezzo and related systems like Placeless show a common underlying metaphor of nonhierarchical associative memory, where coordination semantics can be added alongside content, with an underlying registration model coordinating resources in the case of Intermezzo, all of which I think gets much closer to the goal of “Hmmm…”.

Intermezzo asks us to think about redesignability as an additive process of transformation, where transformations do not necessarily replace or negate previous rules and capacities. Entities do propagate and transform the environment, but in a more fundamentally nondestructive way than is seen in Orbit-style reflection. This may be more in line with a Bakhtinian perspective of reflexivity, where intersubjectivity among many voices constitutes the space of affordances. Intermezzo pushes the redesign capabilities to the designers and their applications operating within the Intermezzo framework, rather than prespecifying a comprehensive reflective architecture. In the Intermezzo approach, adding new semantics and behaviors are additive processes coupled with application-specific code downloaded into the runtime environment that gives elements the potential to be transformative of the system itself. As a whole this appears to be more open than just the three level transformative model of Orbit.

Intermezzo and related approaches might be considered more in line with posthuman understandings of sociotechnical configurations. Generally speaking, Intermezzo doesn’t attempt to impose as strong an ontology based in a humanistic model of interaction and basically conform machines to this, rather it lets the machine and human needs be generated together on a sort of more equal grounds. In this reading, the call to balance human and machine intelligibility
might be the basis for a posthuman reciprocity between humans and machines, and might be extended as a way to represent the exterior agency of technology in constraining or resisting human intention.

In summary, I want to reiterate that Orbit and Intermezzo address somewhat different problems, leading to significant differences in the affordances they provide, particularly when thought of in terms of the problem space articulated in this thesis. Though both overlap in considering the importance of context in structuring a collaboration situation, they address this problem by proposing different infrastructural foci. One might say that Intermezzo is an attempt to create an infrastructure whose goal is sharing coordination information and supporting the development of relational contexts for activity, whereas Orbit attempts to create an infrastructure whose goal is sharing situation particulars in terms of relational contexts, in order to establish a base intelligibility to which coordination is applied. That is, Orbit is an attempt to provide an infrastructure for specifying activities and working together within a structured but open-ended environment, whereas Intermezzo provides an infrastructure for specifying coordination needs within a relatively unstructured environment by default, allowing coordinated entities to develop their own structured activities.

Implications for “Hmmm…” Design

Specific constructs within each system can be related to “Hmmm…”, however for the most part the substantive findings were to resituate “Hmmm…” and consider what of its functionality is covered in existing middleware and what it adds as additional proposed functionality. I will address the resituating of Hmmm… as a middleware approach, then discuss ways in which “Hmmm…” extends beyond the middleware problematic. Next, I will discuss the specific aspects of “Hmmm…” that are left out of traditional middleware approaches. Finally, I
will discuss the possible applications of ideas from the systems considered here to the design of “Hmmm…”

**Resituating “Hmmm…” as middleware.** What I was able to clarify as an outcome of this is the problematic that “Hmmm…” addresses, and the specific solution space that “Hmmm…” proposes relative to existing systems. In particular, it became clear that “Hmmm…” is best situated in the domain of middleware. Edwards, Bellotti, Dey and Newman (2003) point out that middleware has different needs for both design and evaluation than collaborative applications have, and their discussion of these distinct characteristics has clear resonances with the problem space articulated above.

As middleware, the systems considered here take a general goal of accommodating a variety of applications, with their own functions, purposes, and use practices, within a framework that enables rich layered collaboration for situated humans using them. The approaches, particularly in Edwards work, are toward environment-centered support for collaboration in an ecology of components. This is interesting, and has some benefits, particularly the benefit of moving away from a centralized infrastructure and ontology, by virtue of aiming for accommodation of heterogeneity.

Findings of my analysis of Intermezzo, particularly the finding that it is more flexible and powerful than more ontologizing approaches like Orbit, challenges some of the centralization implied in the “Hmmm…” approach as it was outlined in the previous chapter. In this context, it appears that “Hmmm…” was conceived of as too singular, centralizing, and ontological. While the notion of an experiential infrastructure and all the ideas leading “Hmmm…” may still prove useful, they likely will not be useful unless they can be decentralized in critical ways, thus
calling design to focus on a separation of core infrastructure from noncore infrastructure, and
calling more specifically for a way to mediate between applications and infrastructure as
Intermezzo and related approaches begin to provide. This is currently a weak point of the
“Hmmm…” system.

Another finding is that the core features of middleware are a separate focus from
applications built to use it, and thus my approach to analysis may have missed the point, insofar
as it sought to understand experientially the user-intelligible constructs offered by the
middleware. In a sense, the specifics of an interaction is largely a separate matter whose
construction lies largely in the domain of applications, if we take seriously the need to move
away from an ontological approach. In this context, the purpose of the inquiry becomes
analyzing systems in their place within the middleware problematic, as a way of better defining
“Hmmm…” and breaking it down into core features and analyzing technologies in each of those
spaces with a practical orientation.

User-centered design of middleware appears generally feasible, following the guidelines
set forth in Edwards et al. (2003). A particular type of participatory metadesign oriented to
develop an end-user design environment for middleware appears to be a fruitful trajectory for
further research and practice. To situate this reflexively, furthermore, is to acknowledge that
these design contexts, internally reflexive as they may be, are but few among many overlapping
contexts of a more general design environment. Interactions taking place within the ecology of
components and feeding back onto the runtime environment could be facilitated by the
middleware itself. This is the problematic that “Hmmm…” addresses, however, this framing
was not explicit in the earlier articulations of the system.
Resituating “Hmmm…” as metamiddleware. The middleware problematic addresses mechanisms that govern the possible relationships among applications, resources, and infrastructure. The mechanisms that govern how applications, resources, and infrastructure are developed, and how the boundaries of each are defined relative to the combined design-use situation, is the domain of processes that mediate the development and use of middleware. In addition to the middleware problematic, then, there is a metamiddleware problematic. In a sense, what I’m describing is a metacomputation problematic about the situatedness of computation, stated here in terms of acknowledging the implications of a constitutive view of communication. Relevant questions emerge about who gets to define the structures and how these structures come into being, as opposed to simply what they afford in terms of functionality relative to a problem space. Furthermore, this metamiddleware problematic includes concerns of computations being partially determined by situational factors such as users, actions, applications, events, relations, and environment. The role of these factors will be at least partially invisible in the resulting representations of activity. With “Hmmm…” I am calling for modeling this level explicitly, but only partially. Since various types of situational factors are to be modeled and reflectively transform the representations of situations that users are embedded within, we can think of the “Hmmm…” engine as a sort of metamiddleware based in a complex understanding of communication-as-constitutive.

Existing systems as implementation media for a “Hmmm…” prototype. In the variety of approaches to collaboration support environments outlined above, a variety of strategies can be mapped to features and affordances proposed for “Hmmm…”, raising the question of whether these frameworks might serve as implementation media for prototyping this system. However, answering this question definitively at a system or implementation level is
beyond the scope of the current analysis. A second, perhaps more useful way to address this question is to ask not whether existing technologies provide a full implementation infrastructure, but instead whether some functionality of “Hmmm…” can be fit into existing constructs, and thus whether these constructs can be repurposed for prototyping “Hmmm…” functionality. As I discuss below, the effective repurposing of middleware coordination mechanisms to allow a modeling of interaction that is not tied directly to components in the traditional sense, but to the abstract types of structure that are modeled within communication as a material phenomenon. Whether this is the case for any existing system or construct is not clear from the current analysis, due in part to the unique features that differentiate “Hmmm…” from the solutions considered here, however some possibilities emerge that are discussed briefly below.

The locales framework provides a theoretical construct of interaction trajectories which, when applied to all entities in the system, may meet the requirements of temporality proposed for “Hmmm…”, namely, a reflexive relationship between entities across time. Both Orbit and Intermezzo include a layer by which temporal relationships between are directly applied back meaningfully, although it is unclear the extent to which this can be used in real-time to structure complex computations. More knowledge of environment-parsing monitor programs in Intermezzo, and by AOP type reflection more generally, as well as more knowledge of reflective action-process models like that of Introspect, implementations of interaction trajectory and all the other forms of temporal complexity that I’m unaware of, and of the possibilities for metaobject and open implementation style reflection in general, would be valuable to further understand whether these approaches can be used to generally implement temporality at an infrastructural level.
I think a solution like Intermezzo, by virtue of its unique combination of additive as well as transformative ways of configuring the behaviors of resources as well as the runtime environment itself, gives a good potential basis for the underlying infrastructure. On the surface, it appears that Intermezzo’s structure could be used to implement limited prototypes of “Hmmm…”, and with some modifications, might even serve as an infrastructure for deployment of a richer system constituted by a series of dedicated applications extending the underlying infrastructure. In general, Edwards’ implementation of property-based associative entities provides a potential starting point for building the dimensional relationships described as meta aspects in the previous chapter. Scoped and multivalue slots, coupled with extensibility and canonicalization, provide an infrastructure for representing matters of concern being scoped on multiple levels. In this view, referents are negotiated and cooriented around via their attributes and properties, which can be multiple and simultaneously connect referents to different situations, communities, etc. This seems to me directly analogous to the meta aspect proposed in the previous chapter, and the Intermezzo approach as a whole provides a general way to approach implementation of meta layering.

In Intermezzo, a registration ontology structures awareness, session management, and policy. These basic constructs are highly aligned with the proposed solution of “Hmmm…”, though the question remains whether they can be used with abstracted types of components at a more micro level, and whether they can be repurposed to serve a more ontologically generative role as envisioned for them in “Hmmm…”. What’s missing in the current form that these affordances take is their constitution within a responsive stream of interaction, as well as a dynamic feedback loop coming from the environment structuring the stream. Though these are
not explicitly supported in Intermezzo, it appears that the possibility is clearly afforded via the mechanisms of runtime intervention.

**Redesigning “Hmmm…” as (meta)middleware.** This leads back to a consideration of the unique features of “Hmmm…” reinterpreted in terms of middleware. I will consider these unique features below, particularly focusing on how they extend the concept of middleware to address the problematic of meta-middleware. I will conclude this analysis section by reconsidering the design of “Hmmm…” as a new type of middleware that is built upon reflected abstractions of complex responsive coordination mechanisms.

The general middleware approach of treating configuration as constitutive may resonate with understandings of design as configuration work. However when compared to the theoretical depiction of communication-as-constitutive outlined in the previous chapters (to extend to communication-in-design and design-in-communication), the configuration of preexisting elements and components is a limited realization of communicative complexity, insofar as the components and their functionality are already black-boxed and reified before being entered into configurations. While ambiguity, extensibility, and complex notions of identity and equality improve the potential richness of configuration, the infrastructures considered above do not provide a full model of communication-as-constitutive, since only a limited type of communication (articulation of configurations) is involved in specifying structure.

This notion of configuration as constitutive differs from the metamodel approach above, in that communication within these systems, as interaction between participants, can only be constitutive in limited ways, limited to explicit sorts of configuration (i.e. articulation work).
These systems treat explicit configuration and property-manipulation as the constitutive acts, thus bounding coordination and articulation work to explicit forms of specification. There are no direct affordances for implicit dimensions of communicative acts to have structuring effects, as articulation work.

None of the systems considered above deal directly with the particular features of “Hmmm…” that address the unique communication-as-constitutive problematic proposed above, namely intersubjectivity, implicit structuring via a temporal interdependency, and emergence modeling. This provides a rethinking of how to approach a constitutive metamodel—namely, treating fundamentally ambiguous acts of human communication as configurational, and thus constitutive, acts.

The infrastructure of “Hmmm…” is intended to provide the sorts of component flexibility and interoperability that is directly addressed in the approaches considered here, however this is not the central or sole goal of the “Hmmm…” system. As discovered via this process of analysis and reflection, the traditional middleware problematic only partially aligns with the “Hmmm…” system, which has a central aim of also establishing certain dynamics of intersubjectivity within responsive object building and structuring. That is to say, “Hmmm…” aims to provide a game-like experience of responsive co-orientation that governs emergence, where the design of the game can be performed within it. This is not a central focus of the systems considered here, which may possibly support this functionality but focus mainly on straightforward configuration of elements via explicit articulation work.

The technologies considered here are part of a general move to component based systems that do not have a central author, or a central ontology (Dobson & Nixon, 2005). The move to
adaptive systems design perhaps would be a more direct way to engage the considerations of the “Hmmm…” approach. A possible downside of the Intermezzo approach is the inability to relate this middleware itself directly to the level of constitutiveness which is more human-intelligible (i.e. communication as constitutive in the human-intelligible sense), a strength that the locales framework has in theory. Intermezzo relies on a richness of applications to constitute the means and well as the site and purposes of collaboration, and the ostensible reason and telos of interaction are not modeled in the infrastructure. As discussed in Edwards et al (2002), it is the case generally within middleware development that it is hard to get a sense of feasibility without specific applications focused on testing core functionality.

A focus on middleware alone without also considering applications that implement specific use situations and how those come into being, could not address the problem at hand within the reflexive model proposed by “Hmmm…”’. The middleware approaches considered above leave us without a notion of an adaptive semantics that would allow constitutive communication to be directly modeled. That is, the above systems do not directly support the ability to build different communication theories into the structure without adding another set of infrastructure, namely, responsive and dynamic complexity as well as implicit dimensions of communication acts (e.g. encapsulation of utterances as objects, emergence of context specifications from within interaction, etc.).

What “Hmmm…” proposes is a currently centralized solution that would model and render interactive complexity, heterogeneity, multiplicity, etc., among a virtualized or abstract set of communicative entities, where the structural effects of communicative action are in large part implicit and not only derived user-driven explicit articulation work. As suggested above, it
may be possible to break down these functionalities into components that can be overlaid upon an Intermezzo-like infrastructure to provide a prototype of the “Hmmm…” system.

However, I believe that “Hmmm…” would not be most effective as a simulation of dynamics retrofitted to a component-based infrastructure. Because metamiddleware is a different problematic, it calls for a different approach at the level of the middleware itself. Middleware approaches oriented to reified components are limited insofar as they don’t imagine themselves situated relative to an open-ended metamiddleware via which the entities themselves come into being. In communication terms, the issue is providing support at the level of middleware for some coupling with such a metamiddleware system, which consists of a language for interpreting and modeling highly ambiguous intersubjective encapsulations driven by experiential or phenomenological forms of interaction-structuring. To consider this so-called metamiddleware problematic would call these approaches to consider new strategies of interoperability among a broader field of entities, and call for reconfiguration of what constitutes an element from the perspective of the infrastructure, while simultaneously taking some of the core insights and models of their existing middleware approaches to managing coordination, insofar as they are useful, to this broader domain.

**Rethinking the design approach to “Hmmm…”**. We could reconceive “Hmmm…” as an Intermezzo-like schema-less associative memory system, where effective property-specification is not carried out primarily or primordially within a widget, but instead occurs within a real-time responsive layering interface. For practical reasons, this entire responsive middleware structure may be abstracted as a simulation and tied to a less responsive base layer. Further, we can suggest that “Hmmm…” would reflectively couple this infrastructure level interface with a responsive layering interface governing the specification of content elements
themselves, with each treated as separate levels of coordination but otherwise using parallel forms of responsive layering of interaction. These ideas will be explored in more depth below.

At a purely practical level, there is a definite need for separating out the simulated environment where all entities are intersubjective, open, and reconfigurable (e.g. all the responsiveness functionality and everything about the semantic architecture) from a base environment which is more stable as an implementation of coordination, synchronization, data management, platform-interoperability, etc. These two layers would be reflectively connected but not one and the same, and the experienced environment would act as a development environment that could compile changes, that would itself be constituted in a framework that was not as radically reflective as proposed at the outset.

While “Hmmm…” would rely upon a flexible infrastructure for component interoperability like the one proposed by Edwards, its simulation of complex structural dynamics would happen necessarily atop this underlying structure where explicit articulation of components and interoperability occurs; that is, a layer of abstraction up. However the real infrastructure and the communication infrastructure are entangled, so the implementation is not trivial. For this reason, a redesign of coordination infrastructure is called for, that enables the further reflexivity and rich intersubjectivity that “Hmmm…” aims to functionally model.

One basic functional characteristic of the “Hmmm…” approach is a reflexive feedback loop between the outcomes of interaction and the conditions structuring interaction. Rather than seeking systems that are already real-time reflexive feedback loops to the conditions of the system, this study sought systems that had reflective capabilities and real-time capabilities, and that seemed to provide mechanisms for redesign from within the system, as well as offering rich
affordances for structural emergence and complexity. These systems fell short of the real-time reflection architecture proposed for “Hmmm…”; however the analysis did surface some unanticipated alternative approaches to implementing a general feedback loop between structural conditions and outcomes of situated interaction. These general strategies will be repurposed here in redescribing the “Hmmm…” system as a real-time interface to multiple layered abstractions of coordination mechanisms.

I argue that, to address the problem of reflexivity and communication-as-constitutive sufficiently, infrastructure-level support for rich intersubjective renderings of basic elements is needed. Perhaps this is one promise of 3D CVEs that remains unfulfilled, to provide rich and dynamic experiences of the coming-into-being of various forms. A component infrastructure may be used to constitute the actual richness of intersubjective experience from the perspective of the users via some configuration of applications, however it is my argument that an infrastructure-level way to model complex responsiveness is called for in order to accommodate truly reflexive constitutiveness. Complex responsiveness, including notions of interruption, repetition, suspension, multiplicity, replication, and temporary or partial resolution, must be developed at the infrastructure-intelligible level in order to enable the characteristic of intersubjective emergence to be reflexive upon the system itself. This is the truly radical call of the “Hmmm…” approach, that it asks infrastructure, or at least an reflective abstraction of this infrastructure, to constitute coordination based not on transparent encapsulations such as activities, but instead upon a nontransparent engine that dynamically transforms all resources and situates encapsulations in an ongoing process of reconfiguration, repetition, differentiation, and partial, local, temporary resolution.
A rich real-time model of coordination instituted as a way to govern interoperability generally provides the basis for a truly intersubjective and constitutive communication and design ecology. Extending the basic infrastructure-level coordination mechanisms to additional layers of abstraction provides for multiple levels of end-user design. First, it would enable the modeling of different types of coordination affordances and constraints, basically enabling the modeling of a diversity of communicative foundations constrained only by the depth, flexibility and power of the constructs chosen for the infrastructure-level coordination mechanisms. Using these would require a virtualization mechanism whereby new environments with completely different underlying conditions could be initiated from within existing environments. This could be made more accessible by an elegant parameterization of the dynamics of the responsive environment.

Another type of end-user design made possible by further abstracting and building upon the underlying infrastructural mechanisms for coordination is the modeling and codevelopment of dynamic ideas and entities. In this view, users working in a content-development sandbox environment create a variety of distinct elements and explicitly coordinate these relative to one another, and in the process generate temporal objects, sequences or loops of coordinated elements that are themselves encapsulatable (via recording capture mechanisms) as complex dynamic elements sharable and evolvable in the intersubjective ecologies, from which another layer of encapsulation may be abstracted and applied back into the same space, or upon another layer. Such internal coordination within encapsulated elements allows for recursive entities based entirely on the coordination among elements that cannot themselves be tied down to a particular location, as they are themselves involved in various coordination relationships and are defined by this multiplicity of relations.
In a responsively structured system, temporal relations become the main agent of encapsulation following whatever parameters are used to reflexively punctuate the stream of coordination behavior. Multiple, coexisting representations of time, enabling temporal encapsulation while ensuring some degree of coordination and synchronization, would necessarily be included at the level of infrastructure, in order to make intelligible the forms resulting from the complex coordination relationships taken here as the basis of ostensible entities.

Here we can see how reflexively instantiating complex responsive coordination provides the radical possibility of using “Hmmm…” as a reflexive multilayered configuration system. Furthermore, the infrastructure could be conceived and constructed so as to have dimensions of autopoietic communication in Luhmann’s sense of this term, that is, a telos that is limited to creating fields of possibility in relations prompting choice to reduce ambiguity and multiplicity to particular forms. In this way, systems could be structured with their own general telos from these defaults, atop which layers of added semantics and parameterization would be used to structure particular scenarios and enable other sorts of goal-oriented behavior at system or infrastructure level.

The registration relationship described by Smith and utilized by Edwards, becomes more than an objective representation of activity, in the “Hmmm…” system. Registration becomes more closely akin to Smith’s implicitly phenomenological definition, an act of selection that functionally creates basic internally-coordinated elements, populating the s-region, o-region and registration link between them without the content and properties of these subelements being explicitly specified as representing a particular user, object and computational process. However, this is an extension, rather than a dismissal, of Edwards’ use of registration elements.
As in Edwards’ view, a centripetal pull configures multiple pattern-matched registration elements into stabilized or canonicalized subjects, objects, and processes. The core difference is in making some of this canonicalization low-level and implicit from the perspective of the user, residing at the infrastructure-level real-time generation of complex encapsulations.

In addition to making some elements of coordination implicit, there is a separate matter of making some generally invisible elements explicit, though functionally implemented as having a degree of unknowability, undecidability, or randomness. The approach here is to give explicit space for the unknowable via the exteriority meta dimension, so as to signify the invisible excess as a constitutive component of a situation, and allow actors to thematize it and relate around it. I see now that this space for exteriority must be included at the infrastructural level as well, among the various affordances of coordination. This perhaps would allow the strategy taken to deciding the level at which reality is being defined at least partially in terms of what is not being defined; that is, to accommodate an undecidability, as well as multiplicity and temporary suspension of difference, as mechanisms governing the form taken by performed realities.

This, then, is a reconsideration of “Hmmm…” as a sort of metamiddleware encompassing a direct relationship between complex responsive coordination mechanisms and functional encapsulations at the middleware level, and a reflective application of these coordination constructs within user interfaces to provide for direct intervention as well as additive and reconfigurative redesign. I believe that the resulting articulation of “Hmmm…” represents a unique approach to addressing the problematic outlined in the previous chapters, and though untested, makes several specific contributions to designing solutions that extend current middleware perspectives into issues of metamiddleware while allowing these two levels to be
coupled in generative ways. This is the answer to the research questions about feasibility—the findings of the analysis led directly to a reframing of the solution that improves the potential of the “Hmmm…” system to be feasibly developed beyond the naïve articulation specified in Chapter Two, partially by specifying limitations of existing middleware approaches in accommodating theories of constitutive communication, and partially by providing structures by which “Hmmm…” can be concretely defined as an infrastructure.

Conclusion

The technologies analyzed here significantly overlap with the proposed system and point to new ways of thinking about the “Hmmm…” solution space in terms of technological implementation. The approach characterized by Intermezzo reveals a potential infrastructural solution to some of the functionality proposed, while simultaneously challenging the centralization implied in the framing of the “Hmmm…” approach. Furthermore, this analysis guides clearer specification of the unique features of the proposed system. In the next chapter I will reflect on this research process and further implications of the findings discussed above.
Chapter Four

Introduction

This thesis has asked the question whether we can design a technology that serves as a radically redesignable environment for interaction grounded on complex theories of entanglement and constitutive communication. In this chapter I reflect upon the research project and its outcomes. First I discuss whether the research questions were answered, and reflect generally upon the research process. Next, I consider contributions and limitations of the current study. I then discuss implications of the research for theory and practice. Finally, I conclude the chapter by discussing directions for future research.

Reflecting On the Process

Within this research project’s life, there was no shortage of mistakes, blindspots, wrong turns, and general difficulties. Issues that arose within the research process demonstrate the problem of not having a clear enough framing of the problem domain at the outset. The research was characterized by an iterative process of redefining the problem domain along with the solution as further analyses were conducted. This demonstrates something about the nature of design research into wicked problems and has implications for the possibilities of design research in the problem domain outlined above. I will address each of these below.

Answering the research questions. Was the research successful in answering the questions set out at the beginning? I argue that the research project achieved the general aims of reflecting on the problematic and the proposed design approach, however it did not sufficiently answer the question of whether this type of design is actually feasible or not. In this section I will address these questions directly and proceed to consider issues specific to the research
approach, moving to reflect more generally on the implications of my findings about the designerly approach itself.

Do existing technologies demonstrate the feasibility of the proposed system? The systems analyzed did demonstrate numerous parallels in a design space that overlaps with that proposed here, however the research mostly clarified the extent of the problematic rather than concretely commenting on the proposed system. This is both due to the distinction between the proposed system and middleware technologies reviewed, as outlined at the end of the previous chapter, as well as due more generally to issues of framing that structured the entire process and product, as I argue below.

Would it be feasible to actually implement the proposed system in one of the reviewed technological frameworks? As discussed at the end of the last chapter, this is not perhaps even a valid question considering the distinctions between the proposed system and the approaches studied. Because the specific features that embody the unique core of the proposed system were not specified deeply beforehand, this review did not answer that question. If the system had been more clearly defined at the outset, the feasibility of its underlying constructs could have been better tested.

At a basic level, the findings can be interpreted to argue for feasibility in general, yet they also clearly call to the designer for a clearer specification, such as that outlined at the end of the previous chapter, in order to test real feasibility. The architecture of Intermezzo in particular seem promising as approaches to prototype the “Hmmm…” system, however the design articulated in Chapter Two clearly supposes certain important dimensions that are not necessarily included within the aforementioned systems. In any case, in order to validate the Hmmm…"
system by establishing the feasibility of a concrete instantiation, the underlying implementation of the systems analyzed above would have to be better known. That is to say, by backgrounding the nuts and bolts of coordination and synchronization, in particular looking at these in terms of their implementations and the affordances of these for deep redesign within the interface, the research could not assess whether the system was actually feasible.

In the sense that it enabled a better articulation of the problem space, the research was successful, though not successful in developing more specific aspects of this problem space in depth. In terms of testing the specific feasibility of the proposed system, I would say the research was not highly successful. Whether the specific outline of a system proposed in Chapter Two is itself feasible within the parameters of existing technologies is an open question.

**Issues with framing.** One of the persistent (and pernicious) issues throughout was the researcher’s difficulty in enforcing a narrow scope upon the project, which can be at least partially attributed to the overwhelming breadth of the problem domain. This was an issue throughout the various research processes: considering literature that informs the system’s design; proposing the functionality of a system; designing a functionally-specified prototype to demonstrate the functionality; deciding how to assess the proposed design; selecting and analyzing technological frameworks that could tell us something about the proposed design. The relevant research to the problematic was far too vast based on how the problem was initially framed.

In retrospect, the breadth and ambiguity of the problem-domain specification was in some ways a definite mistake, at least insofar as it made the study too large to fit the parameters of an M.A. research project conducted by one individual. This is not simply a practical matter, it
could also be argued to have adverse effects on the intellectual contributions of the study. Without more specific constraints on the research, it was too difficult to make concrete articulations that could truly guide a system’s development. To engage a design problem in a material domain where I did not possess the background expertise nor speak the technical languages of development exacerbated these problems significantly.

This was partly a problem of complexity that was not resolved early on in the process. There are multiple levels on which the theory and design could address the phenomena of communication and design and their relationships to one another. There were multiple levels where theory informed the system, and I acted to summarize and leave most of the detail out in order to fit the parameters of the thesis (e.g. more macro-social and micro-interaction relational theories). There were multiple levels at which a computer system can be analyzed, and moving or selecting among them in terms of framing, performing, and assessing design was too complex for a person without prior expertise in the underlying technologies (hardware, language, data structure, component, application, system, framework, interface, etc.). There was, of course, a great complexity of levels that emerged particularly from the consideration of reflexivity, such as how to place boundaries on a design intended for redesign, which parts of research project should be treated as design itself; at what level to assess outcomes of the research, etc. Without good criteria for reducing these complexities, I was left to attempt to manage them pragmatically, thus limiting the depth with which I could engage any single dimension of the problem or solution space.

From this, insights about the practical importance within design of the act of framing the problem emerged. Many of the ways in which the analysis falls short of offering a strong validation or not of the “Hmmm…” approach can be tied to issues with problem framing.
How the problem is framed plays a material role in what sorts of solutions are considered, and at a more broad level, what sorts of approaches are valued by the professional design communities undertaking work in the problem space. This has implications for the selection process within the review undertaken here, but also for the selection process defining the literature itself. One initial finding was that most of the work on collaboration frameworks that appeared most relevant to the problem as defined above were initially designed and written about in the mid to late 1990s or early 2000s. As mentioned briefly in the previous chapter, the move away from foundational infrastructures for interaction has been documented as a move from second wave to third wave HCI design (Bødker, 2006). In retrospect, it might have been better to conduct a review that was targeted to the later second wave HCI research from the outset, or alternately, a review that focused more on the aspects of “Hmmm…” that align with the more contemporary third wave literature.

My framing was a major limiting factor in the analysis undertaken in this study. Because I was not sure of where to look for the most relevant research, significant blindspots persisted and limited my ability to conduct a robust study. For example, an article by Edwards et al. (2002) about the nature of designing and evaluating middleware provides very helpful advice for anticipating some of the issues I encountered and designing better assessments for middleware technologies, and would likely have changed my entire approach and framing, however it was not part of my earlier literature reviews nor the technology analysis, instead being discovered by me only while already having undertaken and written much of the analysis.

After clarifying the problematic more deeply, a casual inquiry reveals a variety of other technologies that have direct relevance to the problem space which were left out of the current study as outside the selection parameters or not being judged relevant upon initial encounter. In
some cases, relevant approaches may have been overlooked as a result of not matching the problem domain considered above as closely as desired. Alternately, an approach that appeared to match the problem domain well might in retrospect be seen as overdetermining. In retrospect, some criteria were based on limited knowledge, my understanding of which changed considerably throughout.

Further, the way that research questions were operationalized limited the scope of the claims that can be made about “Hmmm…” as a result of this study. In particular, not much can be said about the way that framework development would be crucially determined by the particular interface affordances and constraints of any given implementation of “Hmmm…”.

The difficulty in resolving a tension between metamodel architecture and the experiential application, or between framework and interface, is seen clearly in the potentially misguided attempt to prototype a simplified functional simulation of the “Hmmm…” system in order to assess its feasibility. This process included many hours spent both developing concrete specifications for simplified functional simulations of the rules of the system, and perhaps even more substantially, building a prototype interface and visual representation language that could simulate the system within a pen and paper medium. In retrospect, engaging in this time-intensive process was premature relative to the articulation of the system, and was perhaps misguided in light of the role of infrastructure within the proposed system.

The distinction between the framework underlying the system and the interface-level forms themselves were not sufficiently differentiated and specified to provide for a robust experiment. The system proposed calls for both a rich client interface for developing dynamic objects as well as an underlying framework that provides an environment and ecological conditions for change over time. Prototyping moved the focus too heavily to the client interface,
while the review undertaken in chapter three perhaps moved too far in the other direction, almost completely effacing the experiential dimension of situated users and the affordances of their applications. First, I developed a paper prototype to assess the feasibility of a rich client interface without actually able to test the underlying framework. When this didn’t work, I attempted to assess the feasibility of the framework through reviewing existing frameworks, however I did this without considering the interface of the collaborative application that would instantiate this framework. Clearly, these two parts would have benefited from being more clearly defined, which would have allowed the design and evaluation of both to proceed more productively, and which also would have allowed focused analysis to be undertaken to address core features of one or both parts. Untangling the problem of entanglement to some degree is necessary for design, in order to reentangle these factors via robust design solutions.

**Design research as an approach to wicked problems of (meta)coordination.**

Throughout, the research was characterized by a process of iteration, concurrently developing a definition of the problem space, the articulation of a solution, and a vehicle to assess the feasibility of the proposed approach. In this way, the entire process of research had the characteristics of design research addressing a wicked problem. One outcome, according to Ylriksu et al (2009), of design for wicked problems is to develop a new framing of the problem as a result of the design work. This was the case here, as seen in the resituating of “Hmmm…” as metamiddleware and the specific reconceptualization of its design in these terms. The process was characterized by an ongoing multidirectional feedback relationship between the problem articulation, the proposed solution, the theoretical resources, and the relevant technologies for analysis. The clarification of architectural requirements and separation of concerns within the software happened in the course of the research, not beforehand.
The nature of wicked problems is that they cannot be stated definitively, and that the methodology of a design approach seeks to develop new understandings of the problem domain as a primary outcome. A successful outcome for the research project could be found in both the development of a satisficing solution to the problem of supporting entangled communication and design in computer systems, and the ability to articulate a new understanding of the problem domain as an outcome of this design. In the case of this research project, the wicked problem that drove the research was not entirely well-framed at the outset, and thus to a significant degree the design process was concurrently informed by an iteration upon the framing of the problem space.

In other words, the framing of the research problems was itself an iterative design process; the separation between an initial articulation, a design to address it, and an assessment of the resulting system’s feasibility, and a better articulation of the problem domain was not the result of clear cut stages and separate processes. In general, the process was messy and non-linear, as could be expected for a design inquiry into a wicked problem. This is an implication of the theoretical perspectives on design and communication as iterative sense-making process that cannot be bounded in discrete phases. The written account here is the author’s attempt to simultaneously represent the process as such but use clearer language in framing the analysis, in order to describe the problem and solution spaces in terms that may not have been at my disposal at the time of the initial inquiry.

However, not all of the issues with this research should be attributed positively to the designerly nature of the inquiry and wickedness of the problem domain. Rather, there are some possible implications of the issues encountered about better ways to conduct designer research. I would argue that design research has significant limitations when undertaken alone and directed
towards such a broad problem space. The resources required to be appropriately
interdisciplinary in a broad space are nearly impossible to synthesize, which will likely lead to
significant blindspots within designs themselves. Also, to attempt to design for a problem space
that is highly abstract like this leads to difficulty in explaining your work to potential
collaborators, and makes the practical value of such designs more difficult to discern. Perhaps
this latter obstacle is significant because a macro-level designerly approach is not widely
acknowledged or accepted outside of particular circles within academia, which are themselves
largely separated from one another. That one outcome of this research project is a clearer
articulation of the unique contributions of “Hmmm…” in terms of an established problematic in
computer systems design means that, from the designer’s perspective, this research project was
successful in overcoming some of these translation difficulties.

One aspect that differentiates a design approach from much traditional scholarship in the
field of communication is the orientation to action in the form of satisficing solutions rather than
detailed understanding of the problem space, addressing the phenomena we seek to understand
not only by analyzing, but by “doing something about it”. This appealed to my sensibilities, and
is one of the reasons I chose a designerly approach for this project. However, much of my
training and the disciplinary and genre constraints into which I was attempting to fit this work
called for thorough analysis of the problem domain, which was not possible under the
circumstances. Lacking the expertise of a designer and performing a research project within the
context of an academic discipline where an expository thesis describing a research project was
the necessary outcome, the hybrid designer-academic approach taken here may have watered-

21 e.g. Complex Adaptive Systems and other areas and fields of study influenced by cybernetics, as well as
other places where metadesign ideas have currency.
down the products of both design and intellectual inquiry. This is particularly true not because it’s impossible, but because the author lacked experience in doing this sort of balancing act necessary to produce a robust academic study while honoring the trajectory of design as an intervention into the problem space independent of the analytic project of the thesis. That is to say, this approach of hybridity and the interdisciplinarity of the problem space may have caused the energies of the researcher to be spread too thin, resulting in a less robust design contribution as well as a less robust intellectual contribution.

In many ways, the needs of design and of analysis are competing. The attempt to address both, particularly within a very broad problem space, caused the project to extend far beyond its initial parameters and yield a total product that is less than satisfying as an answer to the specific analytic or design questions about communication and design. At the very least, such inquiry is very difficult and problematic undertaken in the relative isolation that it was undertaken in.

Here as elsewhere we can see the complexity and scale by which the problem articulation can be seen within the life and experience of the research and design process itself. The values underlying the system, namely a relational balancing act that requires as much openness as it does closed structure, should be reflected in the research process, and this insight was not applied as well as it could have been in this project. That is to say, I can’t honestly say my own relational configurations were ideal as I undertook this project within my own lifeworld, and this certainly has invisible influences that can be seen to further limit the power and applicability of the resulting work.
Contributions, Limitations, and Future Directions

What can be seen as veritable problems in the research process did produce valuable side effects, and taken together with the issues outlined above, the study can be argued to make contributions to design research methodology addressing a complex problem space. More generally, some aspects of the research project itself do represent contributions to theory and practice, albeit in limited ways. Limitations to applicability are specifically found in the overall abstraction and effacing of social and political dimensions with this framing of the problem space. These have implications for future directions in design research along the lines of the “Hmmm…” approach. In this section, I will discuss each of the above in turn.

Contributions. While the requirements for interdisciplinarity “spread the researcher thin” in terms of the products of design, I would argue that taking this broad approach had the benefit of exploratory engagement with a much broader field of relevant perspectives. The confusion and difficulty was generative in unexpected ways, and enabled me to come to the conclusions about the both the limits and possibilities of this type of design. While it is beyond the abilities of one researcher to comprehensively articulate the problem domain or to develop a comprehensive solution, I speculate that it may be useful to develop generalists who take a broad approach to design and can synthesize, albeit to a limited degree, a diversity of resources informing a complex problem space. I think also that it’s important to lower the barriers for serious theory to be brought into design practice, particularly with regards to complex phenomena such as communication. This means making it possible for theorists to be designers, as well as acknowledging, following Craig (1999) that designers, insofar as they engage in reflexive metadiscourse, are already theorists. I would argue that a contribution of the current work is to demonstrate the constraints, but also the generative possibilities, for theory-oriented
scholars to conduct design research addressing the implications of their theoretical notions within the field of technology design. While this may be applicable generally, the contribution of the current study is to make this claim particularly around theories of communication, and more narrowly around theorizations of the reflexive relationship of interaction to structure.

If humans hope to metadesign our own social and technical processes at the level of infrastructure, some way is needed to understand and intervene in problem domains that are too complex and ambiguous for definitive sense-making. A possible contribution of the methodology was to demonstrate the fruitfulness of the designerly approach to a complex problem space. It seems that design-minded scholars could adapt to their own needs the general process of synthesizing theory and attempting to naively design from this basis, and subsequently clarifying the feasibility of the proposed design through a review of existing technologies, in order to iteratively shape both the problem and solution articulations in order to produce increasingly realistic and productive spaces inviting further design.

To clarify, I am indeed arguing for the value of a certain designerly naïveté at this broad level of a problem domain. If the starting point had been what is possible within a relatively stable interpretation of existing problem and solution definitions, this type of constitutive metamodel problematic may never have been addressed. In some ways, it may require naïveté in order to create a novel problem and solution definition and to explore creative ways to bring about technical implementation only after the problem and solution domains have been explored in abstract terms. With this approach, I also hope to have demonstrated also that it is possible and fruitful to connect the fields of communication and technology in design, and that undertaking a design approach to research can indeed open up many possibilities for theorizing communication and addressing communication issues. This latter point was not argued here,
since the approach taken did not involve analyzing actual instances of interaction and thus remained in an abstract realm with regard to any claims about real interaction.

The system was functionally underspecified at the outset of this project, and remained underspecified throughout the testing and analysis phases. This was an issue, but may also be seen to have generative capacities when considered in terms of inviting participation. In these terms, the outcomes of the design research were validated in the prototype testing as well as via the analysis of technologies, and we can see a contribution of “Hmm…” as the establishment of a conceptual model that, but virtue of being underdesigned may generate further design efforts into the problematic of reflexivity in communication and design technologies. I will discuss the implications of underdesign and its relationship to participatory practices in more depth below.

After clarifying the the “Hmm…” approach itself in terms of the problem domain of middleware, I can make the general claim that this system makes contributions to the field. “Hmm…” contributes to the design space around collaboration support environments particular conceptualizations of reflexivity and responsivity, outlining unique ways to render a real-time relational emergence and to apply these outputs back upon the functioning of the rendering environment itself. While this thesis does not conclude whether any of the “Hmm…” system elements are completely novel and useful solutions, the analysis of existing technologies did find significant overlap with existing research into collaboration support infrastructure as well as clarifying supplementary features of “Hmm…” relative to the existing collaboration support approaches. By virtue of this, it appears that enough of the proposed solution is indeed novel, and enough of it is actually not novel (and therefore potentially sound, i.e. resonant with approaches taken by professionals in the design field), that an argument could
be made for the likelihood that the design of “Hmmm…” may contribute to future applications in various areas of collaboration technology.

Limitations. Despite the aforementioned breadth of the problem domain, the study had many limitations in terms of how this breadth was limited to a narrower problem space actually addressed by this study. In particular, this study left out questions of the embodied interaction of human participants coconstructing the meaning of the system. That is to say that actual practice was not directly included in assessing the claims, nor was the embodied dimension of the interface considered. This move to abstraction leaves aside questions of culture, power, difference, and other influencing factors, to the detriment of design solutions and the claims that can be made about them within the above framing. I will discuss these briefly, acknowledging that this is only a selective consideration of limitations.

Significantly, one omission from this study was considerations of the most important areas of concern for contemporary communication, technology, and design scholarship alike—culture. By leaving considerations of culture aside, this research lacks the ability to claim value as a generic infrastructure for collaboration for any situated cultural actors. Furthermore, by minimizing the reflexive focus placed on the researcher’s own cultural assumptions, various aspects of the proposed system as well as the technologies considered in relation to it were likely to contain unexamined assumptions that limit their transcultural applicability as ontology for communication and design. Generally speaking, the invisibility of culture throughout was a detriment to the richness of the system design itself and the conclusions drawn from the research.

At the explicitly social level, this research left out critical considerations of power and authority also. The power relations that condition actual communicative practices play a
significant role in all sociotechnical processes, and it is dangerous to ignore this dimension in the specification of a design. By appropriating ideas from theory instrumentally and in simplified forms, designers run the risk of effacing the complexity and richness of human sciences and losing the dialogic spirit of intersubjectivity. The project demands a participatory infrastructure, and this research does not provide a clear model for what such an infrastructure might look like.

In addition to culture and power, there were numerous theoretical and philosophical constructs that were left out of the current study despite having general importance to the underlying design. For example, the research did not consider exteriority, difference, or otherness directly as a framing problematic, and also did not actually consider strategies for exteriority that are posited as foundational to the system design. Strategies for representing exteriority were included in the design as a functional part of the meta infrastructure proposed for micro, meso, and macro levels alike, that could provide deeper intersubjectivity for mediating persistent public material and conceptual matters of concern. Various relational forms specifying both difference and indeterminacy were designed into the proposed system, yet were not considered in terms of their implications within later design and analysis stages. By ignoring this design feature, which could be seen as one unique contribution of the proposed approach, this research further effaced the role of otherness, furthering the separation implied in each of the above limitations of a monologic “design time” authored by a transparently intentional researcher from a presumably intersubjective and dialogic “use time” upon which the system’s very existence is also supposed to be based.
Implications and future directions. It is an open question whether it makes sense to include design considerations addressing these systemic or ecological complexities as explicit concerns within the design of human communication and design technologies—within this thesis it was an assumption taken by the author, however, this is not necessarily in line with the assumptions and values proposed in other literatures. In this section I will consider some implications of the complexity underlying these questions of design practice. Throughout, I will make reference to possible future directions for research building upon the approach taken here.

There is messiness and entanglement in the problem and solution space. Making metatheoretical and practical claims side by side, focusing on the situatedness of some things while effacing that of others, this particular domain is subject to confusion and paradox at many levels. This makes it difficult to develop a common intelligibility to define the boundaries of the problem space. One question that remains open: Considering the essentially interpretive nature of the problem domain, are design inquiries into it generative in valued ways, or do they suggest instead that such a problem domain should not be directly addressed via computer systems design which would by definition overdetermine something that is by definition messy? There is a question that could be addressed to this work as a whole, hinted at above, of whether it even makes sense to attempt to model or simulate a constitutive theory of communication as a solution to problems of reflexively accounting for the effects of communication acts.

As one element of this, it might be said that the proposed design and its general solution space are primarily about retrofitting a constitutive view of communication onto fundamentally transmissional constructs. Indeed, this was the strategy articulated in the previous chapter for re-thinking the “Hmmm…” design as an abstraction from a less flexible and constitutive base architecture. Is this a necessary condition, or a limitation in the thinking of the designer? The
answer is probably that it is some of both. To build upon the current computational infrastructure, it appears to be a necessary condition to use relatively transmissional models of information exchange to govern the implementation of any system. This reinforces the need for a level of abstraction at which the system constitutes fundamental units of ambiguous and intersubjective meaning for users that is at least one step removed from underlying deterministic rules. It is from this level that situations and programs would be built within the system proposed here, and their ability to radically reconfigure the underlying transmissional constructs is one measure of whether they can model constitutive communication. However, there are ways in which the researcher, and perhaps others seeking to address similar problems, can be too quick to move to transmissional interpretations of the desired functionality. There may be a variety of ways to address these problems, and not all of these ways require reduction to the transmissional constructs of current computation paradigms. That is to say, solutions need not be entirely manifested in computational paradigms of transmission, and also that reduction to transmissional constructs need not be problematic if it is not the ultimately determining level at which the system comes to matter. That said, I believe that directly exploring the possibilities for computational expressivity outside the bounds of transmissional communication ontology is important and rich with possibilities for future research within this design space.

A fundamental question within the problem space, for the author at least, regards the possibilities for complexifying our definition of fundamental elements in shared communication spaces, while simultaneously problematizing the absolute nature of relations among entities and their environment, by providing a rich language and interface to comanage processes of transformation and the emergence of structure. While the review dataset provides numerous demonstrations of ingenious approaches to complexifying data and reflexively relating to it, it
appears that researchers have yet to design interventions that aim directly for these underlying goals. This would suggest that we move the focus to designing for incompleteness, multiplicity, responsivity, and provisional suspension of judgment, rather than towards a fully formed solution to address reflexivity directly. This could also be described as design of interfaces and applications to collaborate in meaning-making from the ground up, rather than aiming for formal specification of the ontological relations between objects and metaobjects. The research process demonstrates that a balance between these two approaches is needed.

The review clarified a distinction between infrastructural technologies and those that were involved in ostensibly interacting with and editing the content of the computational environment. Part of the reason for the confusion here is due to what I now see as a particular unique feature of the proposed system, namely, that the interactive elements would also be used reflectively as the language for computer actions that transform the space, as reconceptualized at the end of the previous chapter. Perhaps this calls us to resituate the inquiry to focus first on the feasibility of the first-order model of communication as realized through the rich client interface and infrastructure-level coordination system, rather than on the lowest-level conceptual architecture and implementation. Regardless of where the focus lies, the complexity of the constitutive view of communication offers many questions to be answered by future communication technologies.

In some basic sense, a technology design based in this understanding of constitutive communication and reflexivity questions many of the assumptions underlying design literature. In particular, the design approach here set out to question both the assumption of transparency of designerly practice, as well as the goal of transparent outcomes judged in terms of their effectiveness relative to a singular problem. Each of these is assumptions is addressed in
contemporary design literature; the former in participatory design, the latter in third wave HCI with its focus on designing for ephemerality, playfulness, etc. (e.g. Bødker & Christiansen, 2004). Taken together, this literature questions the underlying orientation to traditional measures of effectiveness that permeate design literature. While the researcher’s ability to transcend these assumptions was very limited, the thesis can nonetheless be seen as something of an exploration into what happens when we question the value of practicality as it is generally framed within design work. That is, when we try to attend to the embeddedness of the designer who is focusing on the practical application of creativity to “make something to solve a problem”, and accommodate the problematization of the designers’ intentionality and framing, we begin to understand design reflexively within messy and responsive realities. It appears that there are interesting ways that this reflexivity emerges within design efforts that are about messy and responsive realities.

There is both an acknowledgment of messy realities here, and significantly, a change in orientation away from reducing or effacing all elements of messiness from within design itself. From the perspective of reflexivity in collaboration that treats design as evolutionary and includes the designer within the design, an approach to design that leaves structures and implementations only partially specified and underdesigned may be an asset or even a necessity. In a sense, the intention of the responsive coordination mechanisms reconsidered as infrastructure is to constitute basic intelligible elements of the system as themselves open and underdesigned. However, at the macro level, the underdesign of the entire system, or aspects of it, may invite more participation in the development of the “Hmmm…” system itself. That is to say, projects addressing this problem space require a deep participatory ethic. To begin collaborative codesign in earnest with a relatively underdesigned system possessing just enough
structure to grasp, and allowing great flexibility in implementation, is perhaps a desirable approach to any larger development of communication infrastructure. That the boundaries are not entirely clear seems fundamental to the model, creating a paradox for intentional design efforts that are singularized. Defined as I have above, the ideas comprising “Hmmm…” are beyond the possible realization of the author working alone, but also in a larger sense are intended to invite ongoing codesign beyond any singular vision, organization of resources, or institution, particularly if such a design is to succeed in forming one basis for a practical metamodel. That is to say that it may be a necessity of development that, even at the earliest stages of the prototyping process, users are able to engage in good faith with the structures presupposed by the system in order to find common ground within them and to transform them.

One implication of this is that we have to be careful when describing in what ways the “Hmmm…” system represents a pragmatic approach to the problem of entanglement between communication and design. In particular, it is important not to confuse the notion of a pragmatic approach to entanglement with a move to simplify or narrow the scope of the problem, moves that are often framed as practical or pragmatic strategies to deal with complexity. My argument is that the problem articulation cannot be reduced, and must be acknowledged as unapologetically wicked, intersubjective, nonsingular, and messy. To take a pragmatic approach to this, my argument goes, is to develop solutions which are themselves situated in messiness, rather than oriented to ordering and reducing the complexity. In a world of transmissional computational systems, we must model and simulate forms of messiness, indeterminacy, and chaos that exist in the communicational lifeworld if we are to pragmatically address the complexity implied by communicative constitution of structure. Of course, the focus on mess is not absolute within a pragmatic approach, but rather is treated as pragmatically important part of
forming a creative balance between imposing order and accommodating transformation within a framework that allows some continuity and common ground. This aforementioned way of pragmatically addressing the complexity in communicative action parallels Edwards’ articulation of a rationale for Intermezzo and other recombinant computing solutions, and it is for this reason that I conclude that Intermezzo offers quite a bit to the reframing of the problem space and further development of a “Hmmm…” solution. In general, one of the outcomes of this research has been developing a better understanding of what a pragmatic approach to this problem domain might look like.

It makes sense, relative to the problem domains being addressed, that existing HCI and CSCW research on middleware selects the more immediately practical goals of making already complex things do work together and adapting through reflection toward system-specific effectiveness measures. Indeed, this is the general design goal of any collaboration or interoperability driven adaptive system (e.g. middleware), however it is not the design goal of a metasystem. Such a metasystem must deal with different measures of effectiveness that accommodate the far more complex metasituation of technology design and interaction around it. While such a system could be prototyped and perhaps implemented on a large scale by utilizing and extending existing implementations of middleware, these technical instantiations would remain only an aspect of the relational system constituting the metatechnology by which different orders of functional reality could come together in “dialogical-dialectical coherence” (Craig, 1999).

I think an implication of this involves a shift in focus to interfaces of experience as well as designed incompleteness or openness in system models of the situation. Seeing these shifts in the context of reflexivity implies that end-user development goals change. In particular, it calls
designers to move away from developing domain-specific EUD tools and instead calls us towards developing basic end-user design interfaces that allow deep participation in constitution processes whereby computer systems (or at least their agent representations) come into being. Centrally, this requires that we establish ontological and architectural conditions that make such a deep participation in constitution actually possible and viable as a language for performing computation and HCI. To signify this affordance within the proposed design was the purpose of what I called initial registration; the goal was to provide a user and system-intelligible way to identify the opening boundaries of subjects and a situation, in order to allow a coordinative process of emergence to structure both the bootstrapping as well as the operation of a full rendering system.

Contemporary HCI research is focused on practical goals that include playfulness, ephemerality, and multiplicity among others and general represent a body of design thinking that responds to the critiques of singularizing interaction frameworks. Within this literature, there is not much talk of communicative infrastructure outside the domain and context-specific type needed for particular types of mediated experience. A playful, dynamic and highly social orientation has not yet been applied back upon infrastructure design in the context of contemporary theory and critique. One possible way to respond to Bødker’s (2006) call for the second and third wave to inform one another may be to develop technologies that provide reflexivity between the action and infrastructure, with the new (fourth wave perhaps?) theme of coproductive performance with (infra)structural implications. The current design project attempts to remedy what I perceive as gap in existing design approaches, in order to better incorporate responsive (infra)structural reflexivity in communication technologies. Hopefully, I
have at least succeeded in providing some ground for future research to explore this problem space.

A drop-off in literature considering architectures and frameworks for collaboration at the generic, one-size-fits-all level corresponds to a devaluing of foundational understandings of interaction in the abstract, in favor of grounded, situated understandings of interaction in specific contexts. By framing the problems in culturally-specific, device-specific, and domain-specific terms, researchers and designers can be more direct in relating their work to the real world. However, the trade-off is that the problem space they are addressing is no longer that of the fundamental reflexivity of local interactions to large-scale social, cultural, symbolic, and material structures. Such structures transcend particular forms of interaction, institutions, and localities to form the substrate that constitutes the broader interdependence of human life within ecological relationships to one another and to the entangled natural, cultural, and technological environment. Without considering how local situations map to the larger relational infrastructures of social existence, HCI design becomes more practical about local and immediate needs, yet less able to practically address long-term needs that involve interdependences outside of the scope of a local context for research and design. That is to say, in the context of interdependence, a local focus along is not capable of addressing evolution, transformation, and the connection between the small-scale and the large-scale, the short-term and the long-term.

However, is development of this sort realistic? A seeming issue within the development here is the actual cycle of development for software, which points to an underlying tension between the exigency of particular needs for technology versus the intentional design of supportive infrastructure of a generic nature. Generally, software develops in response to
various stakeholder-based constraints and ends that lead software to be oriented to direct and immediate benefits within a specific domain. The same is true of the professional academic research cycle and its community of practice. Without some measurable benefit in some specific situation, who would invest their time and energy in developing frameworks at the macro level? How will the ideas contained herein move from conceptual architecture to working applications useful to regular people, particularly since they highlight the importance of reflexivity at the level of infrastructure? The appropriation of these constructs and reframing of them to fit exigent needs will continue to play a central path-dependent role in the development of this and any infrastructure-level system. Implicit in the literature, and very present in conversations with scholars, designers, and nonspecialists trying to understand the proposed framework, is the ethic of “give me a concrete use case”. This is unavoidable even as designers may seek to avoid it. A lot of times usefulness is the gold standard, and the lack of articulated specific usefulness (by virtue of the general problem space it addresses) is seen as a problem.

This says something important about the development of frameworks to address the problem space. It seems to suggest that, were it to be actively developed, short-term utility will inevitably drive the development of technologies that constitute this system to some extent. Perhaps in order to mitigate the corrosive effect this might have on the ability for the component technologies to constitute a generic metamodel, a public and participatory framework may be required to legitimize development of the larger system as a public good in order for it to be viable as such. That is to say that the process of development for such a foundational technology matters greatly, and that in order for it to become useful it must be participatory and open from an early stage, and the development process must be capable of managing frameworks that ensure the interoperability of various code instantiations. In other words, a public institution of
stakeholders must be responsible for managing the fundamental level of code standards as well as integrating the development process.

Many software projects have active user communities, and these are often the most significant sites for the user-centered development of the underlying technology. It would be interesting to consider directly the potential to develop an end-user design infrastructure for “Hmmm…” to facilitate more accessible codevelopment of software. In an open networking paradigm, the possibility for reflective collaborative programming of active environments is already present; the argument that I want to forward here is that, to do justice to the insights of communication studies, we need to begin to treat collaborative programming of active environments, in the most open sense, as the basic infrastructure of interaction.

On a practical level, the research process led me to see the pragmatic value of narrowing the scope and working on specific things individually, if at first only due to the inescapability of a heterogeneous component infrastructure. We live in a world of apps, and any infrastructure for collaboration needs to take this into consideration. But there is another practical reason that coincides with the participatory nature of the project. By focusing on the specification and design of one or another aspect of the proposed functionality (e.g. just the proposed structure of responsivity encapsulated in temporal loops), other uses for these designs may be found, as solutions from one space are mapped onto another by situated designers with different realities. In other words, focusing on the functionality in truly distinct modules may risk the coherence of the system, yet simultaneously will allow for the types of recombination and remixing that the “Hmmm…” system is premised upon. Defining the boundaries between common infrastructure and particular applications and experiences remains an issue with this approach, in particular because it threatens the coherence of a metamodel by virtue of its radical heterogeneity and
incommensurability. The metamodel problematic centers around deciding how, if at all, to orient to designing an infrastructure for dialogical-dialectical coherence without overdetermining the forms of intelligibility that such dialogical-dialectical interaction may draw upon and produce.

**Concluding thoughts.** In terms of the problem articulation, one finding was that the problem space was too broad to be *directly* addressed by a single-infrastructure solution. That is to say, I believe now that the problems of entanglement associated with communication and design are too wide-ranging to develop a clear set of guidelines for the development of a computer infrastructure in the singular. A design approach, like any knowledge production, only comes with partial perspective. This can be seen within attempts to develop frameworks in computer science – the efforts that have sought to engage with this type of domain have ended up offering far more constrained solutions that address certain needs but cannot address all of them (e.g. a rich end-user interaction or design environment, or a framework for building and interactive collaborative and distributed computational resources, but not both). Another difficulty with posing infrastructural solutions is that they have run into practical issues of widespread adoption (e.g. Croquet and Squeak’s lack of adoption among programmers in daily use, and monolithic Orbit infrastructures never making it past a prototyping stage). This limitation is not simply due to a lack of effort or creativity on the part of designers, but rather occurs fundamentally because of the reflexivity of the problem domain—if the entire system is designed for redesign from within, how can it also ensure the provision of rich end-user interaction and design tools?

However, I do not see the paradoxical form that this design problem takes as reason to stop inquiring. First, just because a problem domain is too complex to provide clear, comprehensive, and non-contradictory guidelines does not mean that designing with unclear,
non-comprehensive, and contradictory guidelines is not still desirable. Following STS scholars cited above, we must acknowledge the undecidability of objective truth within knowledge-production in ways that construct, protect and care for certain realities, rather than simply as a deconstructive act. In other words, we must go on practically addressing wicked problems regardless of the epistemological critiques we are faced with, and I believe it is important that designers do not limit their focus to those things which we can know with some certainty and are traditionally safe to address within technology design communities of practice, as much as this ideal may seem fairly unrealistic from within these communities at the current time. I believe that opening up the processes to engage members of design communities with many stakeholders in the design of broad coordination and general expression mechanisms provides a refreshing alternative to the tendency toward ever-narrower and more specific audiences.

The problem space explored in this thesis asks for ad hoc solutions to enable the metadesign of infrastructure. I could also imagine that further exploration of the problem space could yield relatively clear and comprehensive guidelines governing the development of a multiplicity of infrastructures to generally address this problem space. The existing multiplicity of disconnected efforts that effectively address the problem space should not be all that we can expect in the solution domain. Rather, I think that the problem domain can be compellingly articulated and brought into focus through a communication-oriented designed intervention at the level of infrastructure, such that resources to address the problem can be synthesized and extended in beneficial ways for open, participatory democratic processes of sociotechnical figuration. Such a design intervention might usefully be thought of as a metamodel in Craig’s (1999) sense of being applied through the particularity of an overarching practical theory that remains deeply open to transformation and multiplicity.
Conclusion

In this thesis, I made a limited attempt to address the design of a theoretically-driven metamiddleware technology based on constitutive theories of communication. I proposed a system containing a variety of mechanisms intended to model a complex emergence of intersubjective and reflexive meaning through responsive coordination and reflexive layering and redesignability. I analyzed existing technologies addressing similar problem spaces, finding significant overlap between the features of these technologies and the features proposed in “Hmmm…”, while also clarifying the extent to which the “Hmmm…” approach addresses a different problem and solution space than traditional collaboration support. In the process, I developed a clearer specification of the design problems involved in applying constitutive theories of communication in a reflexive way to the development of technological forms of infrastructure.
References


doi:10.1080/03637750701196383


doi:10.1145/1013115.1013151


*Handbook of New Media: Social Shaping and Social Consequences of ICTs* (pp. 163-179).


Suchman, L. (1994a). Do categories have politics? The language/action perspective reconsidered. 
*Proceedings of the third conference on European Conference on Computer-Supported 
Cooperative Work*. 1-14. doi: 10.1007/BF00749015

Cooperative Work (CSCW), 3*(1), 85-95. doi:10.1007/BF01305852


Sumner, T., & Stolze, M. (1997). Evolution, not revolution: Participatory design in the toolbelt era. In 
M. King & L. Mathiassen (Eds.), *Computers and Design in Context*. Cambridge, MA: MIT 
Press.

work within WORLDS. *Proceedings of conference on Organizational computing systems*. 55– 
65. doi:10.1145/224019.224025


