4-11-2008

Designing Tailorable Technologies

Matt Germonprez
Case Western Reserve University

Fred Collopy
Case Western Reserve University

芽|Sprouts Working Papers on Information Systems ISSN 1535-6078

Designing Tailorable Technologies

Matt Germonprez
Case Western Reserve University, USA

Fred Collopy
Case Western Reserve University, USA

Abstract

This paper provides principles for designing tailorable technologies. Tailorable technologies are technologies that are modified by end users in the context of their use and are around us as desktop operating systems, web portals, and mobile telephones. While tailorable technologies provide end users with limitless ways to modify the technology, as designers and researchers we have little understanding of how tailorable technologies are initially designed to support that end-user modification. In this paper, we argue that tailorable technologies are a unique technology type in the same light as group support systems and emergent knowledge support systems. This unique technology type is becoming common and we are forced to reevaluate existing design theory, methods of analysis, and streams of literature. In this paper we present design principles of Gordon Pask, Christopher Alexander, Greg Gargarian, and Kim Madsen to strengthen inquiry into tailorable technologies. We then apply the principles to designing tailorable technologies in order for their design to become more coherent and tractable. We conclude that designers need to build reflective and active design environments and gradients of interactive capabilities in order for technology to be readily modified in the context of its use.

Keywords: Information systems, Tailorable systems, Human/Computer interaction, Information system features, Information systems design

Permanent URL: http://sprouts.aisnet.org/4-7

Reference: Germonprez, M., Collopy, F. (2004). "Designing Tailorable Technologies," Case Western Reserve University, USA . *Sprouts: Working Papers on Information Systems*, 4(7). http://sprouts.aisnet.org/4-7

Designing Tailorable Technologies

Introduction

Tailorable technologies enable end users to select and integrate technology functions in the ongoing creation and recreation of unique information systems. They are used daily as ERP systems, operating system desktops, and word processing software. These technologies are tailorable within the confines of the functionality and components provided by the designers. They also allow for a certain amount of user expressiveness around such things as computing style, program preferences, and aesthetic layout. Technology tailoring has gained increasing importance. Designers have little control over how tailoring occurs as applications move toward user-defined assemblages of distributed, Internet-based services that support the exchange and sharing of data and processes.

Technology tailoring rests on the notion that users ultimately define which functions they use, how those functions are integrated, and how the data provided through those functions is displayed. In today's post-modern world, consumers do not respond to objects like trained workers. Rather, they manipulate objects to accommodate their range of action. "They frequently use the things and services that they buy in unorthodox ways; in ways that are very different from those imagined by the providers, marketers, and sellers – in short, they turn commodities into raw materials for a kind of creative bricolage" (Gabriel 2002, pg. 139). Gabriel argues that the assemblage of parts into a larger whole, or the production of bricolage, is done in a tactical, opportunistic, and *ad hoc* fashion, focusing on short term gain for the bicoleur. Technology has been tailored for years but it is the advent of web services and component architecture that has pushed tailoring to the forefront of our attention for designing systems that support customization by post-modern users.

Several definitions of technology tailoring exist but Morch and Mehandjiev's (2000) definition describes tailoring in its simplest terms. They describe it as the user-defined design of a technology in the context of its use. This suggests that the combinations of available functions are limitless so that even the most visionary technocrat could not predict how they will be configured and used. In our view, these technologies impose additional demands on their designers, because of their high interactivity, their "expressiveness" (Gargarian 1995), their role as "aesthetically potent environments" (Pask 1971), and their ability to support "living patterns" (Alexander 1979) and "metaphor" (Madsen 1989). Tailorable technologies are controlled by users in real time, must support adaptation to the individuals using them, and require the ability to manipulate large numbers of parameters simultaneously.

The architect, Christopher Alexander, calls this a pattern language "that allows its users to create an infinite variety of [artifacts]" (Alexander 1979, pg. 186). Alexander is not the only designer to suggest that artifacts can be built in support of tailorable systems. Gordon Pask, one of the early pioneers of cybernetics, participated in the development of the Musicolour system, a system that took in music and produced colored images. Pask used cybernetics to theorize about the relationship between music and images and related projects aimed at producing aesthetically pleasing environments. Gregory Gargarian created a theory of interactive design in order to explain how good designers at once manage design complexity and promote expressive utility. His work draws on Piaget's (Gruber and Voneche 1977) developmental psychology, as well as notions of situated action found in the work of Lucy Suchman (1987). Finally, Madsen (1989) argued that systems should be designed in support of metaphor. In doing this we create systems that encourage users to continually reflect on the tools that are used on the task at hand.

A unique characteristic of tailorable technologies is their support of two distinct design phases. First is designing the initial, primary, or default state. Prior to the use of any technology, whether tailorable or not, a default state is designed. Second is the act of tailoring, or the user defined design of the technology during its use. User defined designs result in secondary, tertiary, quaternary, etc. states of tailorable technologies. Lévi-Strauss (1966) distinguished between the initial designer of a technology and the tailoring of technology in his discussion of the creation of artifacts from available materials. "The [user] is adept at performing a large number of diverse tasks; but, unlike the [designer], he does not subordinate each of them to the availability of raw materials and tools conceived and procured for the purpose of the project. His universe of instruments is closed and the rules of his game are always to make do with "whatever is at hand," that is to say with a set of tools and materials which is always finite and is also heterogeneous, because what it contains bears no relation to the current project, or indeed to any particular project, but is the contingent result of all the occasions there have been to renew or enrich the stock or to maintain it with the remains of previous constructions or destructions." (Lévi-Strauss 1966 p. 17). The two design phases are also echoed by Hummes and Merialdo (2000) is design-time versus run-time modification of an information system and the user toolbox metaphor used in the design of decision support systems (Turban and Aronson, 2002).

Tailorable technologies represent the apex of this unique, dual-design paradigm. Their initial design is explicit in supporting and promoting user modification. Tailorable technologies are not just *expected* to be modified, they are *intended* to be modified. As tailorable technologies become more pervasive, it is incumbent on information systems researchers to better understand them. It is particularly important to understand how these systems are designed in their default state. The aim of this paper is twofold. First, the technology tailoring literature is used to provide a rich description for identifying tailorable technologies. This description provides a point of entry through which to identify and subsequently study tailorable technology use. Design principles of Pask, Alexander, Madsen, and Gargarian are woven into a technical description to identify the new requirements that are imposed on designers of tailorable technologies, what the challenges in meeting these new requirements are, and how these challenges might be overcome. Second, through a case study in designing tailorable technology, the proposed principles are illustrated and refined.

Tailorable Technologies

A large literature in human computer interaction and Information Systems describes the relationship between human cognition and technology and tells us that users play an integral role in the modification of the technology in the context of its use. MacLean et al. note that it is "impossible to design systems which are appropriate for all users and all situations" (1990, pg. 175). Tailorable technologies represent information systems where end-users actions are not dictated through predefined rules or training on how the technology should be used. Instead, users of tailorable technologies reduce an abstract technology to the specific by altering characteristics of the technology.

Tailorable technologies carry an intentionality by a user in the modification of the technology during its use. This intentionality is constructed around tasks that can be accomplished with the technology, perceived value of the technology, and use of the technology in ways that are similar to past user experiences. With this constructed intentionality, tailorable technologies are not defined only by meeting technological criteria. Users must also be provided with an aesthetically potent environment (Pask 1971) or design space (Alexander 1979; Gargarian 1993) in which technology can be tailored based on user constructed parameters. Alexander describes how users consume functional parts in the production of a larger whole. Through the consumption of these parts, technology takes on the desired states for end users. However, a technology that does not provide the technical functionality or a constructed environment for the consumption of parts will not be tailored (Figure 1).

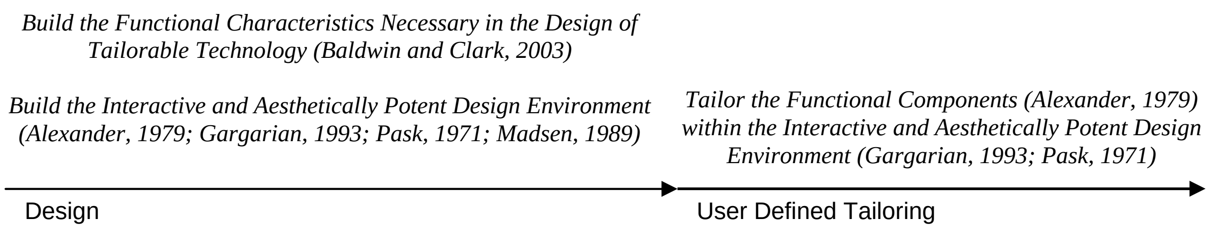


Figure 1. Literature in the design and use of tailorable technologies

Functional Characteristics of Tailorable Technologies

Tailorable technologies are, in part, based on the principles of component architecture where users are able to select from a set of functions during use (Morch and Mehandjiev 2000; Hummes and Merialdo 2000). Component architecture supports user discovery of functions distributed across nodes within a network. At each node, specific, reusable functions can be integrated by users in the formation of unique technologies (Berners-Lee et al., 2003; Baldwin and Clark 2003). The functional characteristics are necessary but not sufficient in designing tailorable technologies, as much of tailoring is based on the consideration and imagination of a user.

Operator	*Provides*
Splitting…	designs and tasks into modules and the creation of new design rules
Substituting…	one module design for another
Augmenting…	is adding a new module to the system
Excluding…	a module from the system
Porting…	a module to another system

Table 1. Operators in the functional design of tailorable systems (adapted from Baldwin and Clark 2003)

Users are constantly integrating smaller, independent components into increasingly complex, integrated systems, in order to manage increasing levels of innovation and growth. Examples of component systems are programmatic subroutines and the object web. What makes component systems unique is that they provide designers an alternative to the decent of designs where systems are subject to a cyclic redesign and a diminishing return model based on improving economic value, system performance, or user acceptance (Baldwin and Clark 2003). Instead, component system design occurs through five operators that supports continual modification and is not subject to designer-centric diminishing returns. The five operators (Table

1) represent the functional support needed for a system to be modified in the context of its use (Baldwin and Clark 2003).

Splitting is used to reduce a single module to smaller parts. The parts may share common features which are aggregated under a hierarchical set of global design rules. For example, a web site can be split into multiple parts based on the data sources represented. Global design rules include consistent format and position. Substitution allows for the replacement of modules or their respective smaller parts based on a value improvement in making a substitution (Baldwin and Clark 2003). Returning to the previous example, if a web site contains a weather data source and a better source is available, it can be substituted for the first. With tailorable technologies, users are capable of surveying a suite of similar and dissimilar modules and substituting modules on an as needed basis.

Tailoring research has explored augmentation and exclusion operations on operating systems (North and Shneiderman 2000), groupware (Wulf et al. 1999), and coordination systems (Cortes 2000). In operating system visualization work by North and Shneiderman (2000) and Dumas and Parsons (1995) “best views” of operating system windows were identified that functionally support user defined modifications. Groupware systems were explored by Wulf et al. (1999), describing how coordinated views are achieved by a group in the ongoing use of a technology. On a smaller scale, Page et al. (1996) looked at the modification of word processing software during its use. For a technology to be modified in the context of its use, it must support the addition and removal of modules.

Using the porting operator, two different systems can use the same module. A module is not bound to a single system but can be replicated across systems. Malone et al. (1995) provide an exemplary piece on porting. Through a series of experiments, the “radically tailorable” tool, OVAL, was used to demonstrate how dependencies between objects could be ported in the creation of multiple information systems that mirror the functionality of the cooperative work systems: gIBIS, Coordinator, Lotus Notes, and Information Lens. Mansfield’s (1997) work on the collaborative system, Orbit, also focused on porting “to offer a deep level of tailoring for groupware [so that] the users have the ability to alter bindings between parts of the system” (pg. 4).

Component architectures are a collection of loosely coupled, independent functions that can be aggregated in the formation of larger systems (Baldwin and Clark 2003). As users perform new tasks, form new groups, or develop new processes, the technology must support these changes (Wang and Haake 2000). As these uses are fundamentally flexible, technology must be able to support this and not strictly represent a set of anticipated user actions. Flexibility relies on a component model and the evolution of component relationships during the ongoing use of a technology (Domingos and Martins 2000; Wang and Haake 2000).

Interactive, Aesthetically Potent Design Environment for Tailorable Technologies

Tailoring will not occur based on the functional characteristics of a technology alone. In order for tailoring to occur, the technology must support modifications in the context of its use and provide an environment that supports and promotes modification. Artifacts, whether a building or a piece of technology, can be architected to encourage modification, producing unforeseen states derived from the original building or technology. The work of Alexander, Pask, Gargarian, and Madsen support a human aspect in the default design of tailorable technologies through the promotion of design environments that supports end user modification.

We concentrate the remainder of this paper on these design environments. We will first explore the key principles of Alexander, Pask, Gargarian, and Madsen. We subsequently reduce recurring themes to a core set of design principles. We then apply these principles to a case in designing tailorable technology. Finally, we reflect the success of the technology and evident design principles in the refinement of the original principles.

Christopher Alexander. Alexander (1979) proposed a design theory that has been applied to architecture, object oriented programming (Booch, 1993), information flows, and organizational design. The theory suggests that systems, whether large or small, architectural or technological, tailorable or rigid, will only be engaging and useful if they are properly defined during their default state. Designing any system starts with a collection of parts and these parts are partially autonomous so that they can adapt to the local conditions (Alexander 1979, p. 163). Similar to Herbert Simon's "artifact," all systems must maintain an inward representation of functional characteristics as well as an outward representation of the context it is being used in. The adoption of singular parts requires a regulation between them to have a specified interaction to design a larger whole. "Design is a process of synthesis, a process of putting together things, a process of combination" (pg. 368) where parts are described first and the whole later. For Alexander, design is ultimately a sequence of increasing complexity where parts are added and the whole emerges.

In the default design of technology, designers must pay attention to what they and their audience know. Designers should imitate existing objects and recognize that these objects describe systems that users ultimately create. Through recognizable objects, users should understand the rules that define their relationships. Like language, words are the objects and rules create the patterns between them. Finally, systems get created by users through the ad hoc, opportunistic, and unpredictable application of rules. It is this final step at which tailoring occurs. In this case, objects are then viewed as complex and potent fields with rules that define their interaction. How those rules are applied represents the tailored system.

To support users in the application of object rules and their ultimate tailoring of systems, designers must present a suite of objects that can be joined for problem setting, not just problem solving. The objects should be analogous to existing technologies in order to promote their use. Objects should illustrate the rules they contain as well as any larger system they are part of. Finally, any object should be functionally complete such that its use provides unique means-end solutions. Each object is then a functional operator which differentiates its own space. It creates distinctions where there were none before. Tailorable systems then become a sequence of these operators.

Gordon Pask. The cybernitician Gordon Pask (1971) observed that individuals search for the novel. When novelty is found, it is explained, solved, or related to an existing body of knowledge. We create metaphors and analogues in reality to search and explain these new environments. The two tasks of searching and explaining describe aesthetic activity and foster social interaction, communication, and cooperation (inner circle of Figure 2). The process of engaging in social interaction occurs when organizing, constructing, interpreting, or appreciating technology (middle circle in Figure 2). Finally, to engage in those activities requires that a person be in an aesthetically potent environment (outer circle of Figure 2).

Sprouts

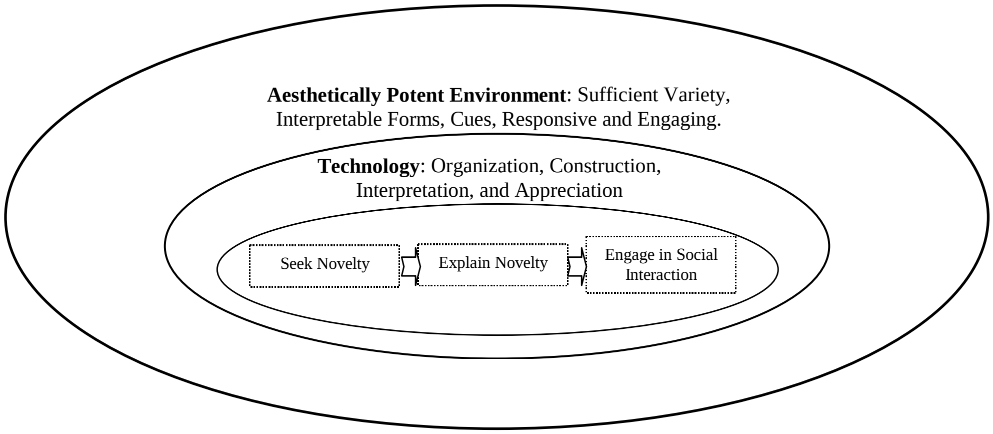


Figure 2. Gordon Pask's model for aesthetically potent environments

Pask hypothesized that aesthetically potent environments share three characteristics. They provide sufficient variety and novelty, so that novelty can be sought. They provide obvious forms and cues that can help explain the newly discovered novelty. They are responsive, engaging a person in actively describing, modifying, and using the technology. The characteristics of Pask's aesthetically potent environment provide a set of principles for designing artifacts that support his notion that, people are always aiming to achieve or discover some goal through technology (Pask 1971, pg. 71). To aid in the achievement and discovery, technology must be naturally engaging, responsive, interpretable, and adaptable. In the Pask sense, tailorable technologies represent any artifact that is used for the explanation and engagement of novelty. It is the responsibility of the initial designers to create an aesthetically potent environment through which the artifact can be used to seek goals, achieve those goals, and engage in cooperation with other individuals.

Greg Gargarian. Gargarian provides a total of 37 hypotheses (Appendix A) towards designing expressive systems. These hypotheses are based on the idea that in any design process, designers must ultimately attend to two aspects of design: the development of the design environment and the production of the artifact that results in the promotion of user engagement and utility. Gargarian proposes three factors that comprise the framework (Figure 3).

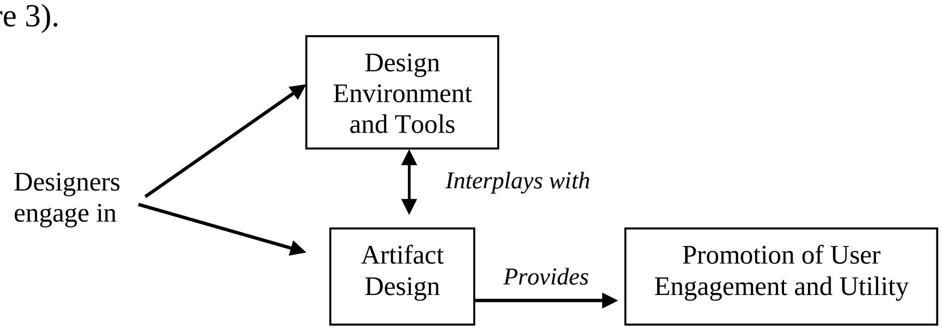


Figure 3. Gargarian's interactive design framework

The design environment provides the tools that enable the production of an artifact, or in our case tailorable technologies. In turn, the tools alter the environment by leading to new ways of thinking. When the environment is altered, *new* tools are identified. These new tools shape a new design environment and so on. Gargarian calls this *learning by designing* and artifacts are produced through the cyclic and discursive relationship. User engagement and utility is built into the artifact based on the interplay between the design environment and the artifact. In order to promote engagement and utility, the artifact must support variety and responsiveness and be composed of features that the user is generally familiar with. Tailoring is then encouraged through recognizable conventions that regulate the ambiguity a user might encounter with the artifact. The Gargarian framework emphasizes a process in designing systems that support, and even promote multiple interpretations of technology being tailored.

Kim Madsen. The final approach comes from Madsen's (1989) work in support of metaphor. Through metaphor, people are capable of understanding one thing in terms of another. Specifically with computer systems, tailorable technologies constantly support metaphor whether through network 'folders,' operating system 'desktops,' or electronic 'mail.' Madsen argues that through metaphor we create and tailor workspaces. That is, metaphor "may be used to perceive a situation in a new way and hereby to provoke invention (pg. 45)."

The process through which metaphor provokes invention is a breakdown. A breakdown moves ready-at-hand technology into present-at-hand. It moves unreflective use into reflective use. It involves the user to create new domains in the use of technology (Madsen, 1989; Winograd and Flores, 1986). Ultimately, different metaphors create different breakdowns which result in technology tailoring. The design of metaphorical systems involves several principles (Table 2).

Metaphorical Design Issue (Madsen, 1989)	*Description*
Structured domains are in advance searched for	Identifying and structuring new domains of use.
New concepts	A technology can be used for any variety of purposes.
Like fiction	Technology is an artifact that inspires new uses
Problem setting	Problems for what the technology can solve are unknown beforehand
Negotiation of concepts	Concepts of how the technology is used are not defined
Several interpretations	Invite many descriptions of the same technology
Incoherence and conflicting	The multiple descriptions may conflict
Analysis and design are integrated	Analysis and design are entirely indistinguishable

Table 2. Metaphorical design issues

Madsen emphasizes a language approach for technology tailoring. In technologies that support metaphor, users can modify technology to formalize new and unanticipated uses, reflect on the use of technology, and restructure their own perceptions of how a particular technology is used. From a technology perspective, metaphorical systems are capable of supporting multiple and conflicting interpretations, and open ended use patterns. Language support in metaphorical technology supports practical involvement and reflection on how particular components can be realized and tailored to support problem solving and negotiated meaning.

Research Model and Questions

From the four approaches, we identified nine recurring factors to provide *a priori* constructs used in our research model. This approach provides grounding for the factors while retaining overall theoretical flexibility (Eisenhardt, 1989). The factors are not necessarily evident in all but present in at least two of the approaches. For example, a recurring factor across all approaches was designing technology that is, analogous to current and in-use systems. Using design approaches of Alexander, Pask, Gargarian, and Madsen, we identified nine factors that contribute to the central outcome of designing tailorable technologies, represent a concept that is generalizable and operational in designing tailorable technologies, and are unique and mutually exclusive. Table 3 illustrates factors that occurred in at least two approaches.

Author	*Alexander*	*Pask*	*Gargarian*
Pask	User Representation		
Gargarian	Recognizable Components Recognizable Conventions Functional Characteristics	Design Tools Design Methods	
Madsen	Outward Representation	Metaphor	Problem Setting

Table 3. Recurring factors between two design approaches

We used the design model of Romme (2003) as a basis for explaining how the factors relate for designing tailorable technologies. The Romme model acts as a way to deductively package the set of factors for describing the design of tailorable technologies. The model comprises a set of factors that collectively portray a unique configuration that defines the purpose of a system, describes its outcomes, and focuses on the development of design theory (Romme, 2003).

The factors represent proposed, not governing principles about designing tailorable technologies. The factors are intended to control the complexity of the design process as well as create usable technology. Specifically, the factors operationalize two design environments: the reflective and the active environments. The reflective environment describes how knowledge and content are used in the *service of action*. The active environment employs the knowledge and content in the *form of action* (Romme, 2003). Table 4 defines the nine factors and their relationship to both the reflective and active environments.

Environment	*Factor*	*Factor Definition*
Reflective	Problem Setting	The technology supports variable tasks and problems.
	Recognizable Components	The technology supports components from existing systems.
	Recognizable Conventions	The technology supports use patterns from existing systems.
	Outward Representation	The technology represents the context which it will likely be used.
	Metaphor	The technology supports symbolic representation.
Active	Tools	The technology relies on existing design tools in its design.
	Method	The technology relies on existing design methods in its design.
	Functional Characteristics	The technology relies on functional requirements.
	User Representation	The technology is designed through representation of users.

Table 4. Nine factors for designing tailorable technologies

We propose the nine factors as a synthetic strategy to process theorizing (Langley, 1999). In a synthetic strategy, the process of designing tailorable technologies is treated as a whole to produce general measures for understanding, explanation, and predictive capability. This approach allows us to avoid describing cause-effect relationships between the factors. Instead, we identify patterns of interaction that can be altered and made actionable for future testing and validation (Romme, 2003).

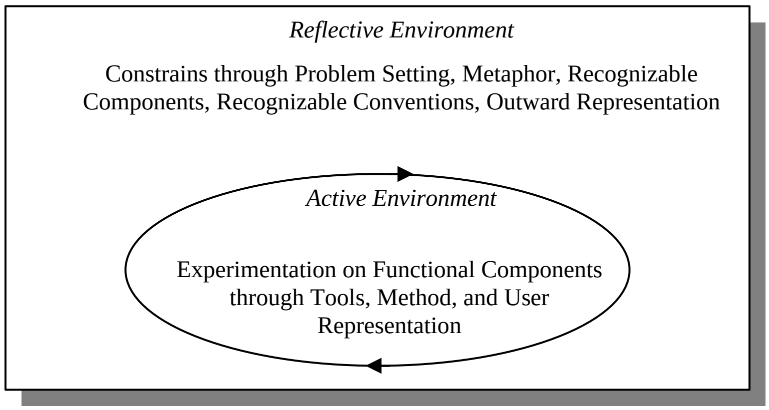


Figure 4. Research model for designing tailorable technologies

In the model both the reflective and active environments contribute in the design and production of tailorable technologies. The approach is process driven where outcomes are future and solution oriented and the reflective environment acts as a set of constraints on the active environment.

In the next section we use a case to gain a better understanding of the presence of each factor and their patterns of interaction. In particular, we use the research model as an assessment tool to define clear definitions and boundaries in designing tailorable technologies at a level that allows comparison among cases (Langley, 1999, pg. 705). Three research questions motivate our theorizing about designing tailorable technologies. First, we ask which of the aforementioned factors are evident in designing tailorable technologies. Second, we ask if any of the factors can be refined beyond their original conceptualization. Finally, what patterns of interaction are present between the factors?

These questions support theorizing from process data of a single case study (Langley, 1999; Eisenhardt, 1989). In particular, they accommodate prior literature in the formation of factors, allow us to gain a familiarity with the data, and sharpen them through the case study.

Research Method and Setting

To answer these three questions, the design of a web portal was studied. The selection of the web portal as a tailorable technology was driven by practical considerations including availability and access to the design team. The web portal represented a fully developed tailorable technology and because our research focused on the design process, we became involved with the design team and test community.

As mentioned, we followed the methodology of Langley (1999) and Eisenhardt (1989) to theorize about designing tailorable technologies. Table 5 represents the issues applied in this research.

This approach has been used repeatedly and successfully from grounded model development of organizational change (Labianca et al., 2002) to designing virtual customer environments (Nambisan, 2002). Like these studies, our purpose is to improve the overall grounding of the factors through prior literature, ground our theorizing through the triangulation of evidence, and build internal consistency by explaining the interactions among factors.

Methodological Issues	*Accomplished Through*
Getting started, selecting a case, and enfolding literature	Definition of a priori constructs in §2 and the selection of a case in designing tailorable technologies.
Crafting instruments and protocols through multiple data collection methods and entering the field	Data collection resulted in a year long qualitative database that included interviews, documentation, and observation from the highlighted boxes. The data collection was often opportunistic to allow for emergent factors and themes. In addition, the first author used the portal for the one year and designed two services for the portal to better understand and verify its functionality. In all, 14 semi-structured interview were conducted, 350 pages of documentation were evidenced, and design of the portal was observed both online and via interactions with the portal manager. Field notes from three researchers were used to orient the data towards answering the three research questions. Interviewees were selected based on their willingness to participate in the research project as well as their position within the university (student, staff, or administration). Of the 14 interviewees, 5 were on-site project designers and 9 were tailorable technology test community members, representing students, staff, and administration.
Analyzing within-case data	Only one case is presented in this paper. This was to accommodate the complexity of proposing a priori factors in addition to a case study. While the multi-case approach was used by Eisenhardt and Bourgeois (1988) and Gersick (1988) in the production of process models, we believe that refinement of our proposed factors and preliminary theorizing was viable through a single case (e.g. Mintzberg and McHugh, 1985).
Sharpening factors to further define, distinguish, and relate	Each factor was revisited multiple times over the course of the data collection and analysis to provide better factor definitions, validity, and measurability. In all, this process provided improved internal validity to the proposed research model.

Table 5. Methodological issues applied (from Eisenhardt, 1989)

Research Setting

Historically, as organizations expand computational capabilities, islands of computing form. Integrating computational islands is a motivator in the development of a web portal. At our case site, the portal was highly integrated with numerous other computing services including email, scheduling, and legacy ERP (Figure 5).

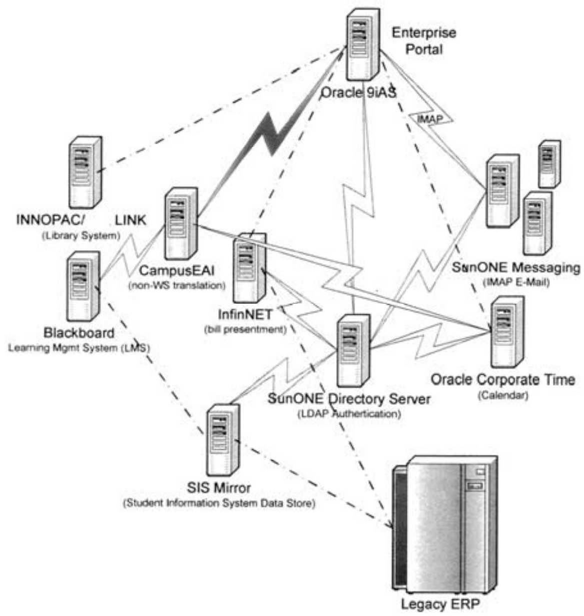


Figure 5. Web portal incorporation of other computing services

Additionally, a web portal provided an interface through which users accessed data in an integrative and personal way. A web portal provides information ranging from the local weather to calendaring functions that can be turned on and turned off by end users (Figure 6).

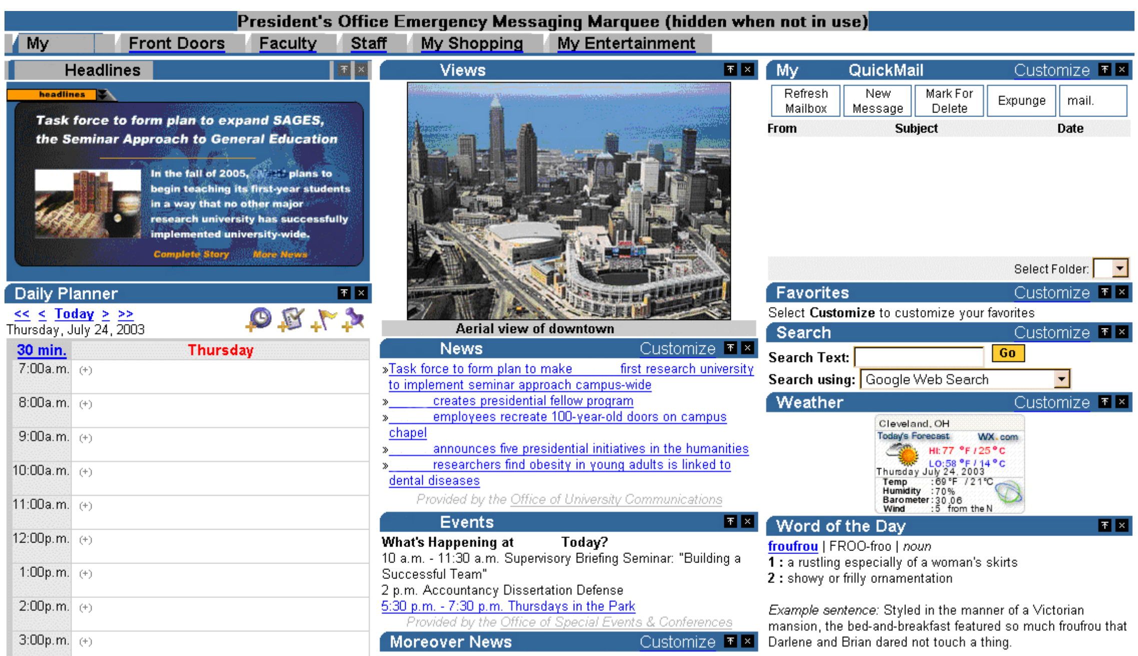


Figure 6. Web portal interface

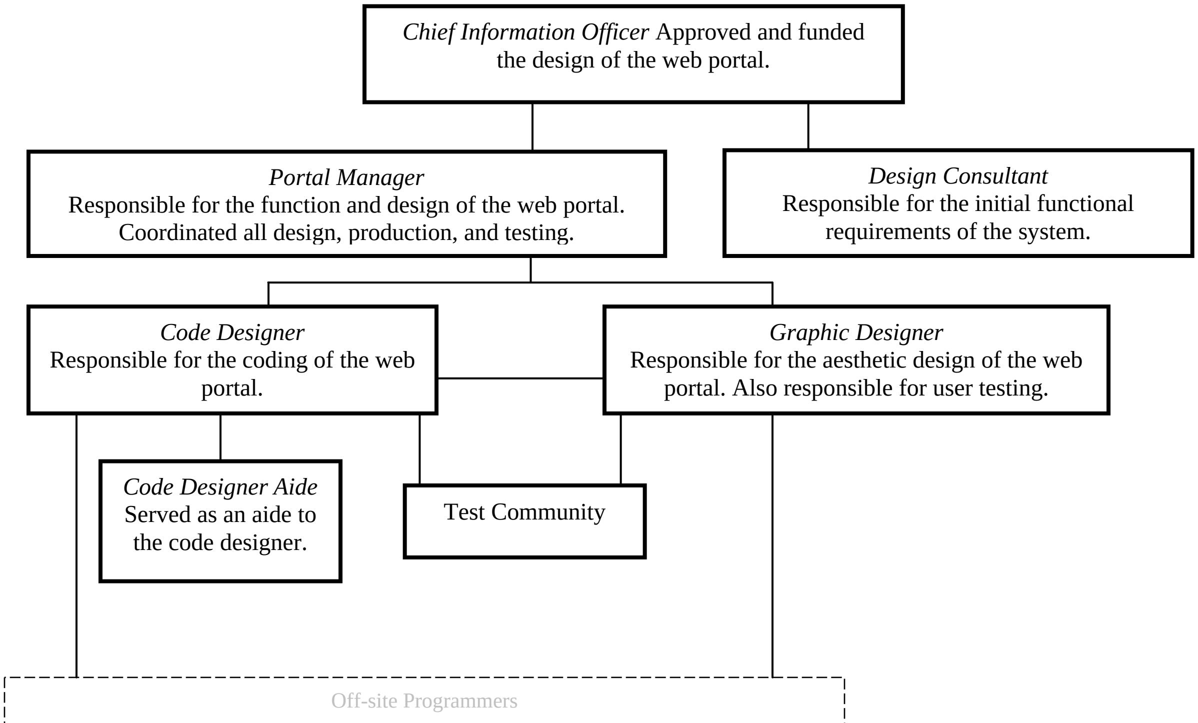


Figure 7. Structure of the Web portal design team

The design team consisted of three administrators, three design team managers, and 20 off-site programmers. The roles are shown in Figure 7. The test community was defined by the tailorable technology project designers and totaled roughly 220 individuals. The test community was identified independent of the research project, based on their association with prior university computing projects, membership in various associations, and employment within university computer support facilities. The test community included undergraduate and graduate students, university staff, and university administration.

Findings

The portal technology was intended to provide a series of 'data peaks' to end users through configurable information portlets ranging from the local news and weather to university-based calendaring and email (D1-I). [1] A goal of the designers was to support unhindered tailoring so users could pick and choose the display and use of any portlet. Restrictions on how users tailored the technology were avoided and the technology was intended to provide anything users demanded, the ability for users to filter any information, and a self-service, user-centric information system (D1-I; D1/3/4-O).

The initial roll out of the technology was considered a working prototype (D2-I). Functionality was gradually increased through the addition of new portlets by the designers.

[1] Support through the case study will be notated as the first letter indicating **D for a member of the Design Team** and **U for a member of the Test/User Community**. Numbering following the D or U indicates different members. Following the hyphen is an **I for Interview**, **O for Observation**, and **D for Documentation**.

Figure 8 shows varying portlets that were added to the system with the first scheduled completion dates and the expected completion dates.

Marketing/Promotion Campaign	9/24/2003	1/9/2004
Pre-Production Hardware Logistics	8/15/2004	TBD
Announcements System		
Tracking Use Statistics		
• in the News* portlet	9/12/2003	Completed
On-the- Agent portlet	9/12/2003	Completed
KSLearn portlet	10/3/2003	Completed
Banner Ad Rotation	1/9/2004	1/9/2004
QuikPay II portlet (aggregate)		
ILLiad (Library) Portlet SSO Access		
SSL Module Rewrite	11/3/2003	Completed
MRTG (Net Stats) Graphs portlet	11/6/2003	11/6/2003
Students First/MyCase Phase I prototype		
Oracle 10g Production Conversion	1/9/2004	1/9/2004

Figure 8. Partial portal development schedule

During the year-long project, all nine proposed factors were observed and their definitions refined. One factor, user representation, was ultimately collapsed into another, outward representation. This was due to the overlap between the two factors and the non-application of user representation in the actual design of the tailorable technology. In the remainder of this section we identify the five factors that comprise the reflective environment, refine them, and then illustrate patterns between them. Following this, we provide the same to the four active environment factors.

Problem setting specifies how a technology could be used. It acts as a fulcrum in the balance between the reflective and active environments for a tailorable technology. The portal technology supported a design split (U1-I) where functionality was designed into the system yet user portals were individually unique (D3-I). The designers rarely prescribed when or how to use the technology (D1-O; D1-D); instead, they provided flexibility (D2-I; U1-I; U2-I; U4-I). Problem setting was accomplished using functional characteristics (U10) and outward representations of the technology to augment spaces where people were otherwise incapable (D3).

Outward representation specifies how the technology represents the context within which it is used. Designers and users recognized that the portal technology could be used to change existing practices and systems into desired ones, even when these ideals are imprecise. The tailorable technology was understood to support changing work practices (U1-I), the evolution of departmental communicative structure (U1-I; U2-I), and cost savings for a department (U6-I). How these are practices were accomplished was less important than the belief that they could be. The tailorable technology was understood as a significant agent for social change, mirroring an existing environment or context and possibly surpassing it (D3-I).

In order for problem setting to occur and outward representations to be recognizable, the technology must support **recognizable components**, or components from existing systems and environments. This factor was assessed through a retrospective look by designers. Each component of the tailorable technology was selected so as to be approachable and usable (D1-D). Recognizable components included communication tools (D3-I; U9-I; U15-I;U1-I), scheduling (U2-I), access to legacy applications (U2-I), and contact management services (U4-I). The portal followed aesthetic conventions of web forms and pages with respect to windows and navigation (D1-I; D1-O).

The technology also supported **recognizable conventions** or use patterns from existing systems. Like recognizable components, this factor was assessed through a retrospective look by designers at the use patterns the technology supports. Generic conventions were employed by the design team based on patterns of conventional web usability (point and click, hyperlinking) (D4/D5/D6-I). The design team provided conventions by designing the technology to support the addition, removal, and rearrangement or portlets similar to other web technologies (D5-I; D1-O). Other conventions included single login (U2-I; D1/D2-I) and repetitive use patterns throughout the technology (D4-I).

Metaphor, and the support of symbolic representation are also evident. Metaphor was present in how the tailorable technology was described, acting as a discursive tool in representing the technology. From a user perspective, the technology was symbolized as desktop like (D1-I), an intelligent agent (D1-I), a marketplace (U6-I), and a communication device (U12-I). From an outward, or contextual perspective, metaphor included the paperless office (U1-I), a tool to reduce organizational silos (D3-I), and a mechanism for porting information from one application to another (U15-I).

These five factors described the reflective environment. The factors were identified and refined, and patterns among the factors are beginning to emerge. Recognizable components and conventions supported problem setting. Problem setting along with the use of metaphor, in turn, enabled users to describe how technology was contextualized and subsequently tailored. Figure 9 shows how they relate.

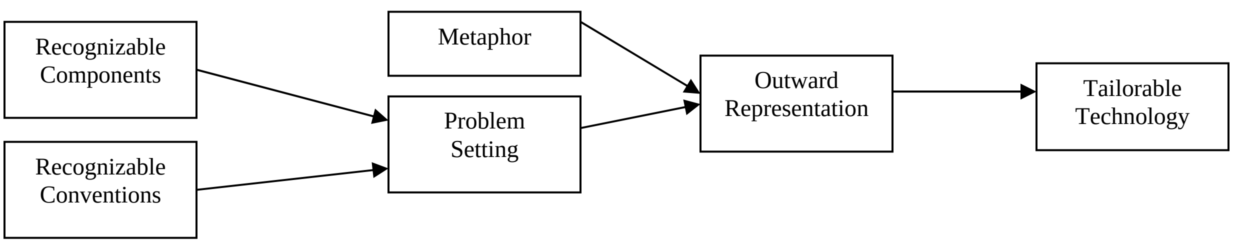


Figure 9. Patterns of interaction between reflective environment factors

With respect to the active environment, tools, method and user representation were all present. Designing the tailorable technology relied on **tools**. The design team used a small portion of the available tools at any one time from a prescriptive software toolbox (D1/D4/D5/D6-O). No formal method for selecting tools was employed; instead they relied on physical proximity of the small team to relate who used which tools for which tasks (D4/5/6-I). The design team learned there were instances where tools lead to new designs which, in turn, lead to new tools and so on (D1/D4/D5/D6-I).

The learning by design **method** was used by the team. Knowing how and when to use tools was always changing. How tools were used and how management styles were shared were informally determined (D1-O). Every designer worked differently and setting common practices or guides for accomplishing work was impractical. The informal approach to sharing common practices pushed each designer to personally select tools, frame their personal design environment, and reevaluate new tools within their own design environment. The management of the design environment was an individual task within the larger group context of producing the tailorable technology (D1/D4/D5/D6-O). The evaluation of when to use tools was informal

(D1/D3-I) and there were no common practices specified (D1/D4/D5/D6-I). Although the tools were prescribed and the method appeared *ad hoc*, neither seemed to hinder designing the tailorable technology. Instead, the design team worked in cycles, focused on knowing functional characteristic outcomes, designing the solution from their tools and method, and repeating this process (D1/D4/D5/D6-O).

There is little doubt that **user representations** should be provided in the design of the technology (U7-I; U9-I; D1-I), that users should be provided training on its use (U1-I; U2-I), and that they should be allowed to provide feedback on the technology (U2-I; D1-I). However, communication between designers and users was limited and users played a marginal role in the actual design of the portal technology.

Finally, the technology adhered to specific **functional characteristics** in support of technical flexibility. The technology provided an integration of legacy systems (D1-D), mandates on certain functions (D1-I) (i.e. presidential banner), and data sharing (U9-I; U15-I; D1-I). The design team treated functional characteristics as the target to which they aimed their design tools and methods. The functional characteristics, in turn, defined the use of new tools and new methods. Identifying, refining, and relating the active environment factors provide our theoretical model (Figure 10).

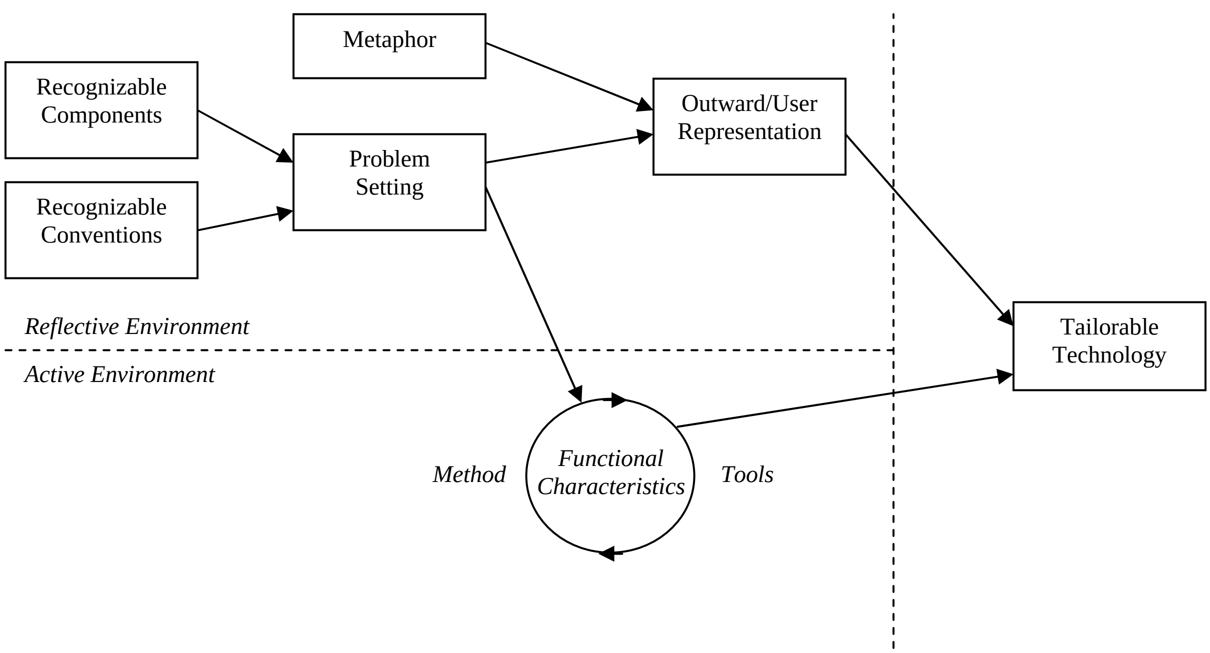


Figure 10. Patterns of interaction between reflective and active environment factors

Designing tailorable technologies is, in fact, two parallel processes with a reflective environment constraining an active environment. The balance between the two environments comes through problem setting, supported by a functionally flexible technology. Through these factors and their patterns of interaction, we theorize about the default design of tailorable technologies. When adopted as a design approach, the reflective and active environments are applied in parallel, and one should not be viewed in isolation from the other.

Conclusions

Our study on designing tailorable technologies offers contributions to scientific knowledge within two areas. The study furthers our understanding of the factors that comprise designing tailorable technologies. This builds on the wide breadth of tailoring literature and specifically on the works of Alexander, Pask, Gargarian, and Madsen and leads to a two dimensional framework for mapping tailoring factors in the design process.

The research also contributes to our understanding of how these factors are related. Following the design of a tailorable technology for a year, it became clear that instead of the design team building a singular, functionally tailorable artifact, they were building a framework upon which tailoring can occur. The team focused on building a capable technology on which users can tailor through the selection, rearrangement, and removal of components based on changing contexts and user expectations. Our model highlights that designing tailorable technologies is the result of two processes: one reflective and the other active.

The findings lead us to wonder whether technology intended to be tailorable is necessarily tailored in practice and if technology that was not intended to be tailorable can be made so through user improvisation. These questions can have an impact on how technology is treated in practice. For instance, if non- tailorable technologies are, in fact, modified in the context of use, how should IT professionals respond? Should they prescribe mandates on use, accommodate the changes through software versioning, or let users tailor freely?

Further studies are needed to develop more specific tactics that can be adopted in designing tailorable technologies. Such efforts should explore the relationship and interaction between factors and environments as well as the mix of research methods needed to study these systems in practice. More effort is needed to contribute to this diversity by strengthening the position of tailorable technologies as a unique information system and by increasing the interaction and collaboration between researchers within our field.

References

Alexander, C. (1979). *The Timeless Way of Building.* New York: Oxford University Press.

Baldwin, C.Y. and Clark, K.B. (2000). Design Rules: The Power of Modularity, MIT Press, Cambridge, MA.

Booch, G., Object Oriented Design with Applications, 2nd ed., Benjamin Cummings, 1993.

Cortes, M. (2000). A coordination language for building collaborative applications, *Computer Supported Cooperative Work,* 9. 5-31.

Dennis, A.R. and Carte, T.A. (1998). Using geographical information systems for decision making: Extending cognitive fit theory to map-based presentations, *Information Systems Research,* 9(2). 194-203.

deSouza, C.S., Barbosa, S., and Silva, S. (2001). Semiotic engineering principles for evaluating end-user programming environments, *Interacting with Computers,* 13. 467-495.

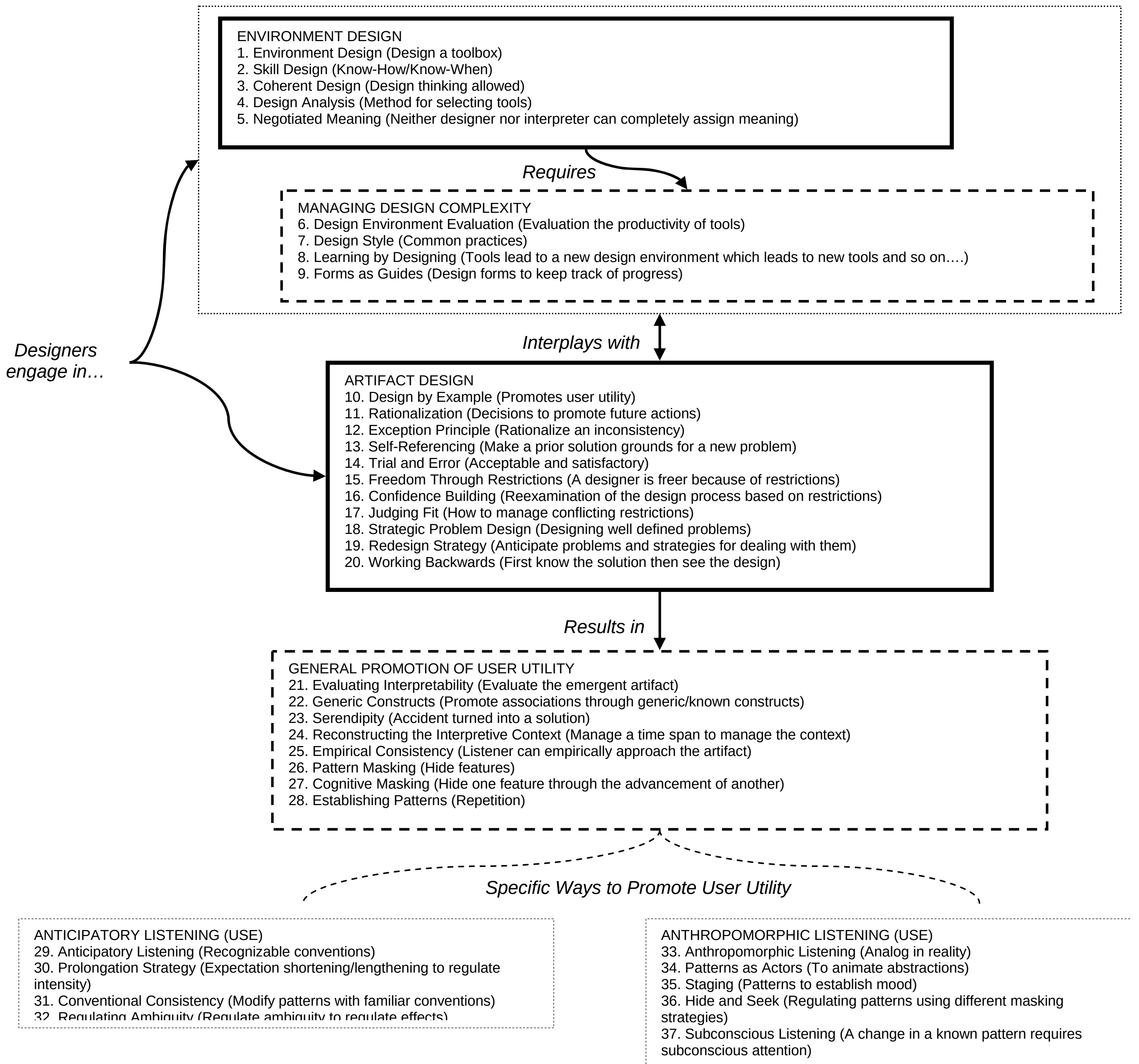
Domingos, H. and Marins, J. (1997). Coordination and tailorability issues in the design of a generic large scale groupware platform, *Proceedings of Group '97,* Phoenix, AZ.

Dourish, P. and Edwards, W.K. (2000). A tale of two toolkits: Relating infrastructure and use in flexible CSCW toolkits, *Computer Supported Cooperative Work,* 9. 33-51.

Dumas, J. and Parsons, P. (1995). Discovering the way programmers think about new programming environments, *Communications of the ACM,* 38, 45-56.
Eisenhardt, K.M. (1989). Building theories from case study research, *Academy of Management Review,* 14(4), 532-550.
Eisenhardt, K.M. and Bourgeois, L.J. (1988). Politics of strategic decision making in high velocity environments: Toward a mid-range theory, *Academy of Management Journal,* 31, 737-770.
Farshchain, B. and Divitini, M. (1997). ICE: A highly tailorable system for building collaboration spaces on the WWW, *Proceedings of Group '97,* Phoenix, AZ.
Fischer, G. and Girgensohn, A. (1990). End-user modifiability in design environments, *Proceedings of the '90 CHI Conference*, Seattle, WA.
Gabriel, Y. (2002). Essai: On Paragrammatic Uses of Organizational Theory – A Provocation. Organization Studies, 23(1): 133-151.
Gargarian, G. (1993). The art of design: Expressive intelligence in music, *Unpublished Doctoral Dissertation*, Massachusetts Institute of Technology.
Gersick, C. (1988). Time and translation in work teams: Toward a new model of group development, *Academy of Management Journal*, 31, 9-41.
Gruber, H.E. & Voneche, J.J. (1977). *The essential Piaget.* New York: Basic Books.
Henderson, A. (1997). Tailoring mechanisms in three research technologies, *Proceedings of Group '97,* Phoenix, AZ.
Hummes, J. and Merialdo, B. (2000). Design and extensible component-based groupware, *Computer Supported Cooperative Work,* 9. 53-74.
Just-Hahn, K. (1997). Tailorable groupware: Issues, methods, and architectures, *Proceedings of Group '97,* Phoenix, AZ.
Labianca, G., Gray, B., Brass, D.J. (2000). A grounded model of organizational schema change during empowerment, *Organization Science,* 11(2). 235-257.
Langley, A. (1999). Strategies for theorizing from process data, *Academy of Management Review*, 24(4), 691-710.
Levi-Strauss, C. (1966) *The Savage Mind.* Oxford, England: Oxford University Press.
Mackay, W. (1990). Patterns of sharing customizable software, *Proceedings of the 1990 CSCW Conference,* Los Angeles, CA.
Mackay, W. (1991). Triggers and barriers to customizing software, *Proceedings of the CHI '91 Conference,* New Orleans, LA.
MacLean, A., Carter, K., Lovstrand, L., and Moran, T. (1990). User tailorable systems: Pressing the issues with buttons, *Proceedings of CHI '90,* New Orleans, LA.
Madsen, K.H. (1989). Breakthrough by breakdown: Metaphors and structured domains, in Systems Development for Human Progress (H.K. Klein and K. Kumar eds.). Elsevier Science Publishers, North-Holland.
Malone, T., Grant, K., Turbak, F., Brobst, S., and Cohen, M. (1987). Intelligent information-sharing systems, *Communications of the ACM*, 30(5). 390-402.
Malone, T., Lai, K., and Fry, C. (1995). Experiments with OVAL: A radically tailorable tool for cooperative work, *ACM Transactions on Information Systems*, 13(2). 177-205.
Mansfield, T. (1997). User-tailorability in a locale-based collaboration system, *Proceedings of Group '97,* Phoenix, AZ.
Markus, L., Majchrzak, A., and Gasser, L. (2002). A design theory for systems that support emergent knowledge processes, *MIS Quarterly*, 26(2). 179-212.

Morch, A. (1995). Three levels of end-user tailoring: Customization, integration, and extension, *Proceedings of the Third Decennial Aarhus Conference,* Aarhus, Denmark.
Morch, A. (1997). Aspects of end user tailoring: People, tools, tailoring, *Proceedings of Group '97,* Phoenix, AZ.
Morch, A. and Mehandjiev, N. (2000). Tailoring as collaboration: The mediating role of multiple representation of application units, *Computer Supported Cooperative Work,* 9. 75-100.
Nambisan, S. (2002). Designing virtual customer environments for new product development: Toward a theory, *Academy of Management Review,* 27(3). 392-413.
North, C. and Shneiderman, B. (2000). Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human-Computer Studies,* 53. 715-739.
Page, S., Johnsgard, T., Albert, U., and Allen, D. (1996). User customization of a word processor, *Proceedings of CHI '96,* Vancouver, British Columbia, Canada.
Reeves, B. and Shipman, F. (1992). Supporting communication between designers and artifact-centered evolving information spaces, *Proceedings of CSCW '92*, Toronto, Ontario.
Romme, A.G.L. (2003). Making a difference: Organization as design, *Organization Science,* 14(5), 558-573.
Steimerling, O. (1997). Using software components as basis for tailorable groupware, *Proceedings of Group '97,* Phoenix, AZ.
Teege, G. (1997). Feature combination: A new approach to tailorable groupware, *Proceedings of the Group '97 Conference,* Phoenix, AZ.
Teege, G. (2000). Users as composers: Parts and features as a basis for tailorability in CSCW systems, *Computer Supported Cooperative Work,* 9. 101-122.
Turban, E. and Aronson, J. (2002). Decision Support Systems and Intelligent Systems 6th Edition, Prentice Hall, Upper Saddle River, NJ.
Wang, W. and Haake, J. (2000). Tailoring groupware: The cooperative hypermedia approach, *Computer Supported Cooperative Work,* 9. 123-146.
Wulf, V. (1997). Tailoring groupware for different scopes of validity, *Proceedings from CHI '97*, Phoenix, AZ.
Wulf, V., Steimerling, O., and Pfeifer, A. (1999). Tailoring groupware for different scopes of validity, *Behaviour and Information Technology,* 18(3). 199-212.

Appendix A: Gargarian Framework for Interactive Design



Sprouts
Working Papers on Information Systems | ISSN 1535-6078

Office:
Sprouts
University of Amsterdam
Roetersstraat 11, Room E 2.74
1018 WB Amsterdam, Netherlands
Email: admin@sprouts.aisnet.org