The Role of Deferred Requirements in a Longitudinal Study of Emailing

Stephen Fickas[1], William Robinson[2], McKay Sohlberg[3]

[1]*Computer Science Department, University of Oregon (fickas@cs.uoregon.edu)*
[2]*Computer Information Systems Department, Georgia State University*
[3]*Communication Disorders and Sciences, University of Oregon (mckay@uoregon.edu)*

Abstract

Our group has taken a clinical approach to doing requirements engineering for a specific domain: delivering email tools to the cognitively impaired population. The clinical view suggests a process that first gathers an individual's goals, assesses the individual's abilities, delivers a tailored system, and finally, monitors usage over time to look for adaptation needs. One concept that has arisen from our project is the notion of a deferred requirement (or deferred goal). Professional clinicians ask an individual to think broadly of the goals they have, and think about not only goals that are achievable today, but ones that might be striven for and become achieved in the future. We discuss this idea of deferred requirements in terms of a longitudinal study working with nine participants over a two-year period. We also report on initial attempts to build automated tools around deferred requirements, monitoring, and system adaptation.

1 Introduction

One view of requirements engineering is that of finding a *reasonable* set of requirements, where reasonable means achievable (a) at system deployment, and (b) with resources available. We would like to argue for another point of view, one that breaks both of these constraints. We are interested in requirements that (a) may not be achievable at deployment time (and, in fact, may never be achievable), and (b) may overrun the set of resources available at the current moment (and forever outrun available resources). We come at this problem from a specific direction. The requirements are ones of *specific users* who happen to have a cognitive impairment. The requirements relate to a user's desires in terms of making use of an email program. The requirements fall under both functional and non-functional classification. The resources available come in two forms: (1) the cognitive resources of an individual user, and (2) the availability of certain types of agents in the environment, e.g., those who can and will act as email correspondents with a user. For the sections that follow, we will focus on a grounded example taken from the email domain to make our points. However, we believe that what we will define shortly as *deferred requirements* have a broader applicability; we will make this case in the closing section.

2 Our Interest is in Clinical Fields

Our study took place in a clinical discipline, where individuals are assessed, individual goals are acquired, each individual is given a tailored treatment package, the effectiveness of the deployed package is tracked for each individual, and mid-course corrections can ensue. Currently, the treatment packages delivered in clinical fields may have a software component, but this software is part of the clinical domain, targeted to professionals providing treatment support. In particular, the clinical software used has little or nothing to do with daily-living software *applications* that many of us take for granted, e.g., email, web browsers, digital photography tools, music-management tools, travel-planning tools. The authors, as part of a multi-disciplinary team, took on a project to make modern software applications available to the cognitively impaired population. This paper is based on a Brownfield approach, where a requirements engineering process was integrated with an existing and well-established cognitive rehabilitation process. One outcome of this integration effort was the emergence of a new type of requirement, one that is forward looking and future-based. We eventually came to think of these requirements as deferred: they were collected at analysis time but their actual achievement was deferred until a more suitable context. In the next sections, we will use a longitudinal study to ground discussion.

3 Cognitive Rehabilitation is a Clinical Discipline

In the spring of 2000 we began a project with an interest in assistive technology for the brain-injured population. The problem the group wanted to tackle was the social isolation a person encounters after a brain injury. The proposal was to give the population access to email and community travel to help break this isolation. Before discussing this specific problem, we will provide some background on the brain-injury population and the discipline of cognitive rehabilitation.

The description of the Mayo Clinic's MIPCR program (see the sidebar) provides many of the high-level pieces of the cognitive rehabilitation field. In particular, it lists the general impairments that may be encountered with a brain-injured individual. It highlights the high-level, *individual* process of assessment, goal-setting, creation of treatment plan and then periodic monitoring. We would add to this the following background information.

- The term clinical centers on the *individual* diagnosis and treatment of *outpatients*. Cognitive rehabilitation fits this definition: it works with individuals and attempts to integrate them back into their communities.
- The size of the brain-injury population is roughly 6-10 million in the U.S.[1].
- The impairments suffered can be broken into two broad classes, memory and executive function. Impairments associated with memory are in the areas of new learning and short-term memory. Impairments associated with executive function include issues with attention, problem solving, and self-monitoring.
- There is a range of severity on all impairments. The combination and severity of impairments varies so that it is not possible to build a "typical" member profile of the population.
- The life goals and interests of individuals vary widely, just as they do in the non brain-injured population.

Mayo Interdisciplinary Program for Cognitive Rehabilitation (MIPCR)

People who enter the Mayo Interdisciplinary Program for Cognitive Rehabilitation may have problems with:

- Attention and concentration
- Memory
- Organization
- Problem solving
- Other cognitive skills that affect life management

Participants must be able to identify thinking problems and set treatment goals.

Before they can enter the program, candidates must go through an evaluation. A neuropsychologist reviews the candidate's medical history and referral. A neuropsychological evaluation may be conducted if one has not been completed recently.

Participants meet individually with an occupational therapist to set treatment goals and create a treatment plan. Additional services can come from a speech pathologist, recreational therapist, neuropsychologist, physical therapist, physiatrist and vocational and community resources.

These people and other professionals meet regularly to monitor the participant's progress. Family members also may take part in treatment sessions.

The length and number of treatment sessions is based on the participant's progress in meeting treatment goals.

http://www.mayoclinic.org/physicalmedicine-rst/mipcr.html

What is missing from the MIPCR description is a reference to what is known as Assistive Technology (or AT). The use of AT will be found if one digs a little deeper into the activities of the cognitive rehabilitation staff listed in the MIPCR program. For instance, one or more of the speech pathologist, recreational therapist, neuropsychologist, physical therapist, or physiatrist may recommend some assistive device that can help an individual reach a goal. It is worth differentiating between non-computer AT and computer-based AT. Non-computer AT has a successful record of accomplishment, e.g., memory aids such as DayTimers are ubiquitous in the population and are viewed as highly effective. Computer-based AT devices have a less stellar record. Abandonment of such devices has been reported as high [9]. While there have been no formal studies on the system-abandonment topic that we are aware of, informally professionals have pointed to two causes: (1) lack of initial fit of an AT device with an individual's needs, and (2) lack of device adaptation once an individual's needs change[9].

We are now ready to turn to the study problem: mitigating the social-isolation that comes with a brain injury. Our interdisciplinary team includes (a) computer scientists, (b) those doing research and practice in clinical cognitive rehabilitation, (c) those doing research into AT training, and (d) those focusing on qualitative research and fieldwork. The team chose to look at email as a technology that could open lost channels of communication with family and friends, and potentially open new channels with some rudimentary pen pal services. The general study question was can email be made a treatment in a rehabilitation clinic looking at social-isolation issues. The following plan of attack was created:

1. Conduct focus groups to get a general view of the population's use of computers and their interest in using email. (Eventually 85 people were interviewed [17]).
2. Build prototype email clients and conduct preliminary usability studies [16].
3. Put together an RE process that fit with the cognitive rehabilitation field. In essence, view

email as a type of AT and integrate it with current clinical practice.

4. Select brain-injured participants. Because of the gradual change in the population, and because of the long-term abandonment issue, we choose a longitudinal approach, working with each participant for at least a year.

As a consequence of the focus groups (step 1), we found that participants were (a) universally excited about the prospect of using email, and (b) highly pessimistic that they could use computers because of past failed attempts[17]. Our pilot usability studies (step 2), revealed that an email client with roughly three parameters of 15+ settings each could be made to work for the participants we studied[16]. This leaves us with the heart of the study, steps 3 and 4: defining a clinical process that delivers a software system (an email client) and measures its success over an interesting time frame. For the latter, in particular, we chose to conduct a long-term longitudinal study that allowed us to monitor changes over a 1-2 year period.

4 The Definition of a Clinical RE Process for Email

Our longitudinal participant group consists of nine brain-injured individuals. Three live in their own homes, four live in an assisted living facility with minimal supervision (they are free to come and go as they please), and two live in an assisted-living facility with high supervision. As we go through the steps in the RE process we have developed, we will present examples from our participant group to ground discussion. A more general view of the project can be found at www.think-and-link.org.

4.1 A Goal Attainment Scale

The cognitive rehabilitation field uses a goal attainment scale to acquire the individual goals and desires of a person. Each goal is broken into a set of attainment levels to provide a measure of attainment. For instance, a goal might be to be able to do shopping for meals, with this broken into degrees of attainment/satisfaction, e.g., can shop for all meals, can shop for special meals, *etc*.

Using this style, we attempted to capture the goals of an individual in terms of email use. (We will henceforth use the terms goal and requirement interchangeably.) We asked each participant to first list a goal and then five levels of attainment, ranging from not-attained to fully-attained. Several examples might help illustrate its use. (We will change the names of individuals for privacy.) First, we introduce Mary. One of Mary's goals is to write letters to the editor of an online newsletter that runs articles on the brain-injury population. She divided this goal into the following categories:

Level 1 (not attained): will not be able to learn how to use email.
Level 2: will be able to write email opinions to friends.
Level 3: will be able to write an email opinion, that meets stylistic constraints, as a submission to the editor.
Level 4: will have a letter printed in the newsletter.
Level 5 (fully attained): will be invited to write a guest letter or opinion.

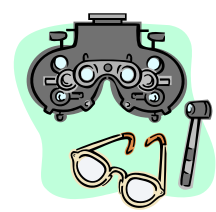
Another example is that of Don. One of his goals was to learn to email with no help. He divided this goal as follows:

Level 1 (not attained): will not be able to learn how to use email.
Level 2: can email, but only with lots of prompting and help.
Level 3: can email, with some prompting and help.
Level 4: can email with no prompting and help.
Level 5 (fully attained): can teach others how to email.

Each of the nine participants typically listed 4-5 goals. It is important to note the temporal nature of these goals, and in particular, the influence the division into attainment levels has on thinking of milestones. No participant expected to fully attain his or her goals from day one. Instead, they were set as achievement targets. This is the norm in clinical fields: individuals state long-term goals and then work towards them.

4.2 An Assessment of Abilities

Members of the team felt strongly about the concept of functional assessment established in the cognitive rehabilitation field. The general idea is that you cannot get a true picture of an individual's abilities in using an AT device by giving them abstract tests. You need to test them using the real device in as realistic setting as possible. In our case, this translated into assessing their ability to write email. We developed an assessment tool built on top of our parameterized email client. Much like an optometrist will go through a series of differential tests to obtain a picture of someone's ocular ability,



we developed an assessment process that "turned" the parameters of our email system to get the best fit with an individual's ability to click around an interface, read directions, write a composition, remember the task at hand, and stay focused. A snapshot of one setting of the interface is shown in Figure 1. This setting is on the power-user side of the scale, having minimal prompting and a split inbox that includes old and new mail. It also contains ten "email buddies" (all part of the research team), some of which can only be accessed by scrolling the inbox. None of our nine participants was able to use this interface effectively during assessment. However, two are currently using it now (after roughly 12 months of emailing).

The outcome of assessment is an inventory of the abilities of an individual related to email. We developed an email-skill inventory that included more than 50 separate skills. Each was rated on a three-point scale of missing/sporadic/possesses[16].

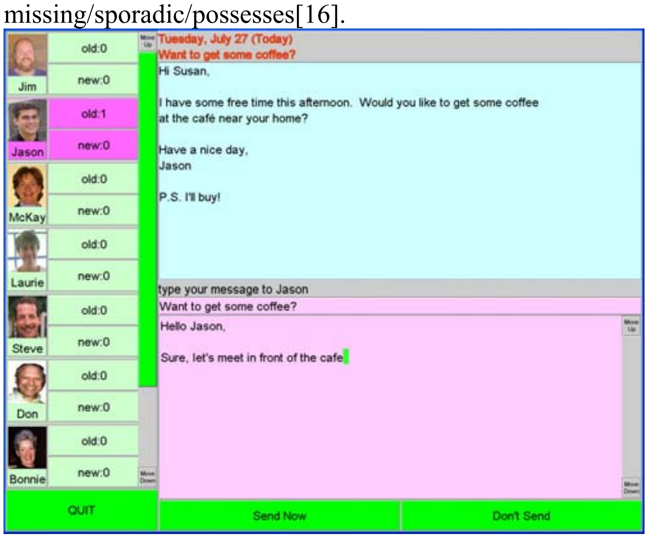


Figure 1 An example of a custom email client.

4.3 Providing a First System

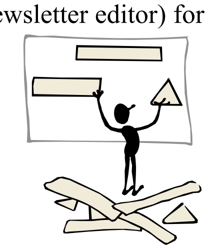
At this point, we have a set of goals, broken into levels, and a skill inventory. For each goal, we derive a set of enabling skills. If these skills are present, then the goal is made active. If one or more of the enabling skills are lacking, the goal is marked as deferred. We will illustrate this shortly.

The set of active goals is used to tailor the first delivered system. Beyond a set of look-and-feel settings, there are three parameters we are concerned with: (1) the setup of the buddy list (the social space), (2) the management of the help system including prompts and reminders (GUI), and (3) specification of the email process the system will enforce (business rules). The actual delivery of the system is a straightforward setting of parameters to instantiate a new email client. That is, we have developed an email client that allows us to add and remove features, and reconfigure the GUI, based on a set of XML specifications. Our parameter setting is actually a reconfiguration of the XML specification.

4.4 Monitoring and Adaptation

The clinic-based view of treatment in cognitive rehabilitation uses goal attainment scales, assessment, and then delivery of a treatment package that is individualized. It also has a built-in step of continuous monitoring of progress, through regular follow-up sessions. In our view, this process has a direct tie with what we call goal-directed monitoring: use the individual's deferred goals to decide what usage behavior is important to track. Using this idea, we developed a type of requirements monitoring, based on a mixture of the goal-attainment-scale (see section 4.1) and our knowledge of what abilities were prerequisites to goals. We derived sub-goals from the different levels of each goal. We viewed these sub-goals as milestones (or stepping-stones) towards full attainment of a goal. An example might be useful here. Looking at Don's goal in section 4.1, he has a desire to be completely independent in his emailing, needing no help from the system (or humans). After completing an assessment (see section 4.3), it turns out Don will need a system with a full array of prompting and help. However, we will attach resource prerequisites to each deferred sub-goal and monitor for the satisfaction of those pre-requisites. When they are met (if ever), we can consider the adaptations we will need to make to Don's client to bring his deferred goal to the forefront. For instance, once he shows proficiency with a portion of the emailing process, the actual adaptation taken will be to remove some piece of supporting system scaffolding.

Mary illustrates somewhat the inverse of the adaptation done for Don: Don had something removed; Mary will have something added. In Mary's case, she wants to send opinion letters to the editor. We divided Mary's goal into sub-goals by using the attainment levels she provided. There are pre-requisites on each of these sub-goals, some tied to use of the email client and some tied to the constraints (placed by the newsletter editor) for accepted submissions. One example is that of length: a letter to the editor must not be longer than twenty lines. In monitoring Mary's correspondence with friends and family, she is initially observed to write very long emails, all well over 20 lines in length. However, with coaching from others, she is observed over time to write a wider variety of email, some short notes and some longer stories. The ability to write shorter size email marks the acquisition of a prerequisite that is attached to her sub-goal of writing a legal submission. When all prerequisites are met, we can mark the sub-goal as active, moving it from the deferred state. This often triggers an adaptation to the system. In this case, the adaptation is to add the editor of the newsletter as a correspondent, i.e., modify the buddy list. This does not achieve her deferred goal of having one of her submissions accepted for publication, but does add another key prerequisite: the ability to submit online (the only form that is acceptable to the newsletter). She can now start submitting letters to the editor.



4.5 The Outcome

The process we describe in this section has been used successfully for all nine participants. While none has fully achieved all of their goals, no goals remain below level 3 for any individual. With few exceptions, each individual started with their first level goals. They then progressed through a set of deferred goals and associated system adaptations.

From the cognitive rehabilitation viewpoint, the project has been highly successful. As noted, virtually none of our focus group participants thought they would be able to use a computer, let alone be accomplished emailers. Further, by adapting to changing needs, we have so far avoided the curse of computer-based AT, that of device abandonment. In summary, the clinicians on our project now believe that our email tool can find a home in the cognitive rehabilitation process[17].

From a software engineering point of view, we have moved the software development process to within the clinical framework where the team is most comfortable. This took modifications. An individual assessment process had to be established. A means of capturing individual goals, and linking deferred goals to monitoring and adaptation had to be defined. We have honed the process over several years to a routine methodology, allowing new participants to be brought into the project and moved smoothly through the steps.

5 A Look at Automation

The RE process described in the preceding section is largely manual. Deferred goals, for example, are recorded in the clinical notes for each individual and then revisited on a weekly basis. Raw email-usage data is studied (by project staff) to determine if deferred goals can be brought to the forefront. At the same time, adaptations to the email client are considered.

Although technical adaptations of the email client are relatively easy to accomplish, decisions about when and what to alter in the application (i.e., redesign) require careful consideration of collected data. Statistical and qualitative data, along with direct input from the CI user, are reviewed at weekly staff meetings to determine whether the user is ready to take on new goals. These are manual processes, which do not scale beyond small studies. In this section, we will discuss steps we are considering to automate portions of this manual process. Our conjecture is that automation may be one approach to the scale issue: perhaps the machine can take over some components of monitoring and adaptation-suggestions that now requires manual effort.

5.1 The Designed Artifact

Over several years, and nine different users, we have developed an email client based on an adaptable, component-based architecture. The system is a highly parameterized, dynamically configurable, event-listener architecture, Java-based system.

The email-client design space for our user population is modest. Using the rather small number of parameters settings, we do not expect the design space to grow much beyond 5000 configurations. While the size of the design space is tolerable, finding the right combinations for any specific user (i.e., the right point in the space), at the right time in their long-term system use, is critical.

5.2 Data Collection

Once an initial system is delivered, the user begins emailing and producing usage data. The form of this data is as follows:

1. Raw data is collected on system usage. The system is built to collect usage data at a fine grain. Every user action is logged, including mouse movement on the screen as well as errors in using the interface, e.g., clicking on an item that has no effect in the current state.
2. Project staff periodically check in with the user to calibrate goals, skills, and general feelings of accomplishment.
3. Online questionnaires are sent periodically to a user's email buddies. These elicit information about the user's emails skills from the buddies' perspective.
4. Changes in the user's ecology are logged. These can include a change in living arrangements, changes in the user's medical condition (including changes in medication), and changes in the user's social circle.

The email client collects a great deal of quantitative data, whereas the qualitative data is collected through online questionnaires and forms. Both kinds of data are stored in our server.

5.3 Goal Specification

We have built a monitoring system that automatically links raw data on system usage to primitive properties associated with deferred goals. In essence, we look for the occurrence of primitive property satisfaction, and then mark associated goals with their appropriate degree of satisfaction[4,5,13,15]. Our real-time monitoring system, called REQMON, has two modes of operation[19].

1. *Model-based monitoring*: using properties derived from user goals, REQMON accumulates information that indicates degrees of goal satisfaction.
2. *Pattern-based monitoring*: using data mining techniques, REQMON accumulates information about reoccurring patterns.

REQMON is configured to raise alerts, or execute a program with changes in property satisfaction. It monitors deferred goals and initiates adaptation as deferred goals becomes active.

As developers, we recursively refine the user goals into a conjunction of sub-goals until primitive monitorable properties are specified. The KAOS Objectiver tool supports this goal refinement process (www.objectiver.com) [2].

Although we demonstrate KAOS goals herein, Fickas and the Toronto OME team have collaborated to define an OME version of the email goals[8]. Together, the two approaches will eventually test the generality of the monitoring and adapting system.

To illustrate KAOS goal specification, consider typical user emailing goals. They will include subgoals for the basic steps of emailing. For example, consider the following simple goals, which are included in our common goal library.

$G_{presence}$: The period between viewings of the email in-box shall be no more than k days.

G_{send}: After composing an email message, the user shall send the email, within k hours.

G_{read}: After noticing a new email, a user shall read the email, within k hours.

G_{reply}: After receiving an email, a user shall read and reply to the sender, within k days.

$G_{read-delete}$: Before deleting an email, a user shall read the email.

Clinicians want to see: (1) a good success-to-failure ratio over sessions, and (2) a constant or improving trend of this ratio. This leads us to refine these goals further. For example, the preceding G_{reply} goal has the following two refined goals:

$G_{reply-ratio}$: The ratio of successes vs. attempts for email replies shall be ≥ 75%, with any two-week period.

$G_{reply-ratio-trend}$: The trend of goal $G_{reply-ratio}$ shall be positive (increasing), with any two-week period.

These goals must be defined in the REQMON language to enable their monitoring. An analyst manually applies property patterns to define the REQMON properties. The Dwyer temporal patterns, implemented in REQMON, simplify this activity. The Dwyer patterns define scopes, including `global`, `before R`, `after Q`, `between Q and R`, and `after Q until R`. Property patterns include `universal`, `absence`, `existence`, `bounded existence`, `response`, `precedence`, `chained precedence`, and `chained response`. The patterns have LTL (and other) formal definitions[3].

As an illustration, consider formalizing (in KAOS) and implementing (in REQMON) the preceding G_{reply} goal.

```
Goal Ideal [Reply]
 UnderResponsibility u1 //Mary
 FormalDef
  ∀ m1,m2:EmailMessage, u1,u2:User
  EmailArrive(m1,u2,u1) ⇒ ◊<t EmailSend(m2,u1,u2)
```

```
(monitor
 (name reply-to-buddy)
 (property email-read-send-sequence))
(sequence-property  ;; G-reply
 (name email-read-send-sequence)
 (scope in-session-scope)     ;defined elsewhere
 (constraints ?*same-buddy*) ;defined elsewhere
 (patterns ("read" "send"))
 (timeouts (reply-timeout))) ;defined elsewhere
```

The KAOS goal `Reply` represents our informal G_{reply} goal. The KAOS language makes use of real-time temporal operators, including: some time in the future (◊), the next state (○), the previous state (●), some time in the past (◆), always in the past (■),always in the future (□), always in the future unless ($\mathcal{W}$), and always in the future until ($\mathcal{U}$). The language also makes use of the standard logical connectives. To address real-time constraints, temporal operators—including $◊_{\leq d}$ and $□_{\leq d}$—have deadlines, *d*, which bound property satisfaction. The temporal operators have common definitions.

The preceding monitor, and its referenced property, specifies how `Reply` should be monitored by our Jess-based[8] implementation of REQMON[14]. The KAOS goal represents an instance of the Dwyer `responds` pattern, which is implemented with the REQMON `sequence` property pattern, a generalization of `responds`.

The preceding monitor specification is typical of Dwyer properties. The monitored property is divided into a scope and a base property. The base property specifies the pattern for matching events. The pattern elements “read” and “send” are referenced in the `same-buddy` constraint, as the following shows.

```
;; A portion of the Event constraints…
(Event (?eventId)
 (EmailEvent (id ?eventId)
 ;; Match the event type with the nth pattern
 (event ?eventType
    &:(nth-pattern-matchp
                              ?eventType ?n ?patterns)))
```

This portion of the constraint matches only email events (constrained by `?eventType`) according to the nth pattern; that is, “read” first and “send” second. The remaining portion of the constraint, not shown, ensures the appropriate correlation among the users, emails, the read-send order, and the send timeout.

It is important to note that monitors can publish their evaluation events into the event stream. Consequently, monitors can analyze monitor evaluations, including their own. In particular, monitors can analyze property ratios and trends, such a required by goals $G_{reply-ratio}$, $G_{reply-ratio-trend}$.

5.4 Goal Monitoring

A monitor incrementally evaluates its property specification as it acquires events. A specification is fully evaluated when an event sequence can be shown to satisfy or violate the specification. A specification is partially evaluated after matching an event and before its satisfaction is determined.

Consider monitoring the preceding `Reply` goal. It is satisfied by the abbreviated event sequence presented in Table 1. For example, the `Session.start` and `Session.end` events *open* and *close* the `in-session-scope` scope, respectively. While its scope is open, the monitor checks its base property specification against arriving events. The `read` and `send` event sequence satisfy the monitor's base `sequence` property. The monitor's satisfaction is recorded when its property is satisfied within an open scope. When properties are violated—for example, because of scope or property timeouts—the violation causes are recorded.

Table 1 Incremental evaluation of events by the reply-to-buddy monitor.

Event	*Resulting evaluation*
E_1:Session.start *Monitor scope opened*	PropertyEvaluation(pe$_1$,in-session-open,(E$_1$),satisfied) ScopeEvaluation(se$_1$,in-session,open,pe$_1$)
E_2:Read E_3:Send *(to E_2.from)* *Monitor sequence property satisfied*	PropertyEvaluation(pe$_2$,email-read-send-sequence,(E$_2$),patial) PropertyEvaluation (pe$_2$,email-read-send-sequence,(E$_2$,E$_3$),satisfied)
E_4:Session.end *Monitor scope closed, and its scoped property evaluated*	PropertyEvaluation(pe$_3$,in-session-close,(E$_4$),satisfied) ScopeEvaluation(se$_1$,in-session,open,pe$_1$,close,pe$_3$) MonitorEvaluation(reply-to-buddy,se$_1$,pe$_2$,satisfied)

5.5 Monitoring Component Architecture

The REQMON architecture is comprised of five main components.

- EVENT CAPTURE listens to event sources, such as the email client events, to monitor system behaviors.
- REPOSITORY maintains the SQL database of events and property evaluation histories.
- ANALYZER updates the status of monitors through event stream property checking. Monitor specifications are compiled into (Jess) rules, which work with the kernel rules to implement the property checking algorithm[14].
- PRESENTER maintains user displays of events, properties, and monitors.
- REACTOR does actions in response to REPOSITORY changes, such as property violations. Actions can include reconfiguration of the targeted software components.

Overall, the system works like a real-time rule-based system, with working memory serving as an event cache.

As an illustration of how REQMON operates, consider the `Reply` goal. If a user does not send a reply to a read email within the specified timeout, then a `Reply` violation is raised and recorded. When a violation occurs, compensation rules execute in response, for example, reconfiguring the email system to prompt for an email reply. It is these rules that enable the types of adaptations discussed next.

5.6 Adaptation

Our focus in this paper is on deferred goals. However, deferred goals clearly have a link to adaptation: using Objectiver, a user's goals are captured, those goals are refined into subgoals until skills are derived that can be monitored. This entire structure is used to support adaptation. Adaptations are called for when resources change in a way that affect a user's goals. These resources can be those of the user, e.g., the user's email skills, the user's ability to physically access a computer. Resources might also involve the user's ecology, e.g., the availability of email buddies, the availability of a computer. Changes can be for the positive – a deferred goal can now be made active – or for the negative – an active goal is lost and moved to the deferred state.

In our email client, there are three adaptation points that we have at our disposal: (1) adaptations to the social space, i.e., the buddy list, (2) adaptations to the GUI, and (3) adaptations to the process/business rules that govern the control of email steps. Given our interest in automation in this section, the question is what support we can give a careprovider or staff person in making adaptation decisions. The discussion in the last several sections on formalizing goals and linking them to monitoring goes a long way towards giving us automated adaptation: if we can map usage events into monitors and then skills, and finally to goals, we can use the goal graph itself to suggest necessary adaptations. For instance, Mary had three key adaptation points in the drive to meet her write-letters-to-the-editor goal:

1. A change to her GUI that allowed her more freedom to type free-form letters.
2. A change to the process rules which allowed her more freedom to save drafts of letters.

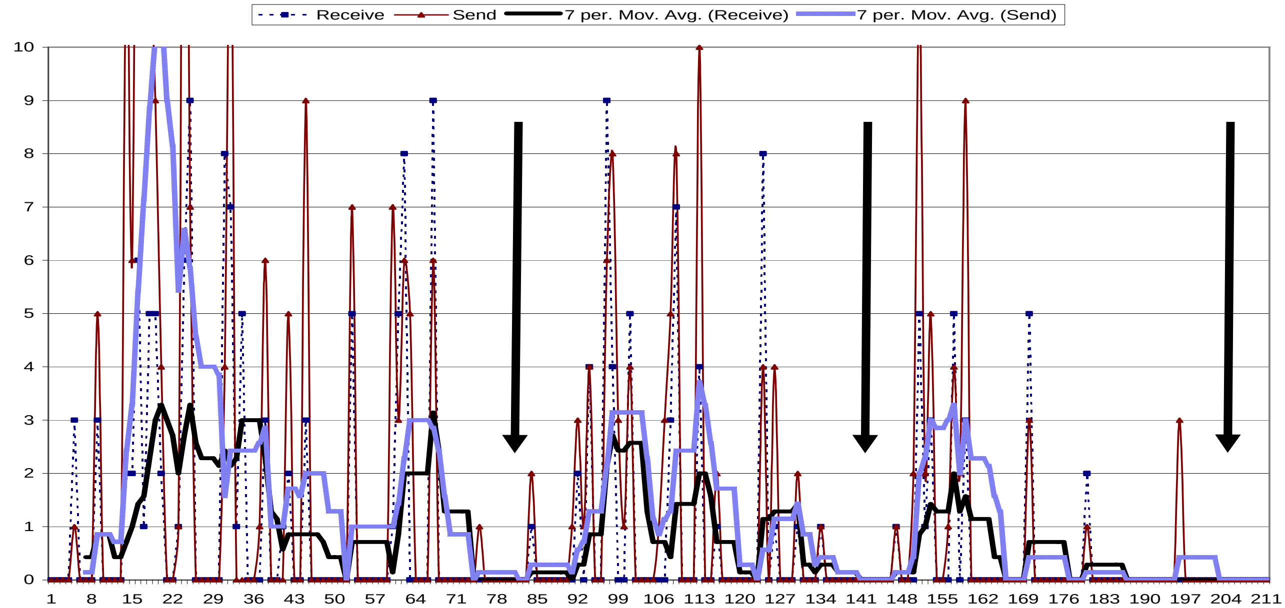


Figure 2 The first 210 days for Michal showing 7 day moving averages of receiving (solid line) and sending email; data points are shown behind the trend lines. Arrows point to periods of email client adaptation using new "email buddies."

3. A change to the social space in the form of adding an editor buddy where she could send her letters.

In each case, we have been able to use Objectiver and ReqMon to link the adaptation to mid-to-low-level goals that represent prerequisites to her larger goal. In summary, if one can know ahead of time the structure of the goal graph for each user, then there are means of automatically traversing it to find adaptations when tied to monitors like ReqMon.

We are also interested in cases where we do not know the entire goal graph a priori. Perhaps it is a new goal that we have not yet encountered. Perhaps it is difficult to derive a general structure that works with large numbers of users. In these cases we may need to learn adaptation rules. An example drawn from our study is that of Michal. One of Michal's goals was to have a large social circle. We know that the adaptation that supports this goal is adapting the social space, i.e., adding a new buddy. The question is what skills can we monitor that are linked to this adaptation. In Mary's case, the link was well defined: add the editor buddy when Mary had prerequisite skills in writing letters-to-the-editor. In Michal's case, adding a buddy to increase ones social circle is idiosyncratic and difficult to link to general skills: when to make changes to ones social circle is fairly personal. We decided to run a small experiment to explore what type of automation support we might provide in Michal's more knotty adaptation problem. Our general approach was to do a retrospective study, using REQMON to analyze data for the number of emails sent and received each day by Michal for the first 210 days. Figure 2 illustrates two things. First, it shows Michal's email send and receive numbers (y axis). Second, it shows where buddies were actually added by project staff (down-pointing arrows). What project staff learned over time is that Michal did best when new buddies were introduced in lulls of activity. The question we are concerned with is what would have happened to Michal if an entire research team had not been available to analyze his usage data. Instead, perhaps only a single (overworked) careprovider is available. Our interest is in using this careprovider's time in a judicious fashion. We conjecture that the machine could have found the correlation between lulls and adding buddies (after observing the careprovider for some period), and used this to take the careprovider out of the data analysis role. With some form of graphic visualization such as figure 2, we conjecture that the careprovider can take on the role of decision maker: the machine proposes an adaptation based on correlations it has found; the careprovider says yea or nay. We are currently exploring these conjectures by using straightforward data mining techniques built-in to ReqMon to produce proposed adaptations, and allowing careproviders to judge their accuracy.

6 Related Work

Group-level requirements are familiar in RE methods, e.g., SCENIC [11], Volere [12]. However, change over time is not explicitly modeled apart from concerns over requirements creep and evolution.

In terms of adapting a system over time, the field of human computer interaction (HCI) has made a distinction between generic task support requirements (i.e., functional requirements) and system features which can be customized for individual user needs [10], [19]. Two architectural approaches to handling personal requirements are adaptive systems in which the system monitors the user's behavior and then changes services or the interface look and feel to match the user's needs; and adaptable systems where customization is user-controlled [6]. In essence, the email client we constructed can be viewed as an adaptive system under these definitions. However, the HCI work does not draw in the notion of deferred goals, actually founded as goal-attainment scales in the field of clinical rehabilitation. Also, the look-and-feel and interface interests of the HCI work were generally *not* the major adaptation concerns in our study. Instead, we found ourselves focusing on the remaining two adaptation points: process/business rules and the social space.

The areas of adaptable software architectures and product lines both are concerned with building systems that can evolve over time. Clearly, our email software shares this concern. However, we have not found the actual construction of an adaptable system to be the problem, but instead *how* to adapt it. The knotty problem is linking system functionality to (deferred) goals of an individual: given goal G of user U, what changes to U's system will support G? Complicating the question is the state of resources available to U, e.g., what skills does U currently possess, and how do they influence changes to the system?

Special Education has well-established (and legally mandated) procedures for working with children with special needs in the K-12 setting. The field has several processes in place that fit with the personalized view of a problem. In the US, the most highly accepted and standardized process is the IEP: Individualized Education Plan. Currently the IEP focuses on the educational components of a child's school experience. The IEP assessment is not dissimilar in intent to our notion of assessment. The IEP process works to tailor an individual education plan for the student that fits with his or her needs. The tie to our work was brought home to us when the director of a special education program for young adults contacted us about adding our email client into the assistive technology available for his kids.

Finally, our own group is attempting to reuse the RE process described for email on a new rehabilitation application, a trip-planner that aids a person with a brain injury in making small community trips. We are just now in the set-up phase of this work, laying out skills, goals, trip-planner software, and monitoring and adaptation structures for community travel [18].

7 Conclusion

We will first summarize our study, and then discuss its larger issues. Our objective was to introduce a software application, email, into an existing cognitive rehabilitation clinic. Directly, this led to two major shifts in our thinking: (1) we needed a *personal* RE process that worked with an individual's requirements, and (2) we needed to work with the concept of *goal-attainment scale* prevalent in the rehabilitation world. Indirectly, we also found the need to follow a clinical process that includes a treatment cycle of monitor-adapt. Using a longitudinal study, we showed success in using a modified RE process that includes an individual assessment step, a goal-attainment step and a monitor-adapt cycle. In terms of this success, the clinicians on the project were astounded by the results obtained: people with no hope of using a computer starting the study were competent emailers within 6 months. And just as important, none had abandoned their system at the time of this writing.

One of the issues with our work is our study methodology. Because our longitudinal-study approach is qualitative and within-subject [17], we cannot say anything definitive about the *necessity* of using the RE process we derived; given there were no large N group-comparisons or control groups we can only say it was sufficient to obtain results. However, we can strongly speculate that any RE approach that does not take a personal view of each user, and capture their long-range goals, would not have a fit with a clinical rehabilitation process. If, like us, one believes that the place to start to deliver software to the cognitively impaired population is by working with rehabilitation specialists in existing clinics, then a process that captures the very personal point of view endemic in such clinics is essential.

The focus of this paper has been on the concept of individual goals/requirements captured in a way that allows their staging over time. Of course, it is fruitless to gather deferred goals if they cannot be profitably used in the treatment cycle. We discussed our fledgling work to formalize goals and automate portions of the treatment cycle. This work is motivated by scale issues: we are concerned that the staff-hours required to run our study will not be available to other clinics. We are exploring automation is one potential solution, reducing staff time to decision making as opposed to detailed analysis.

Finally, we have considered whether the personal RE process we defined, with its inclusion of deferred goals and a monitor-adapt cycle, can have any advantage in a non-clinical setting? We will take an example to ground discussion. One of the authors, Fickas, is an amateur photographer. He has a goal of showing some of his work. He is reasonably proficient in the (physical) darkroom, and has submitted some of his photographs to local art shows. However, he found that transferring his physical darkroom skills to a digital darkroom has been a challenge. In essence, he is Photoshop impaired.

Fickas claims that he would pay for a service that would treat him personally. He would like a service that would (a) assess his digital imaging skills, (b) pay attention to his individual goals, (c) break those goals out using the goal-attainment scale concept, and (d) monitor and adapt a digital imaging tool over time. First, what would be required for this service to even function?

- The service would need a theory of digital-imaging skills to back-up an assessment step.
- The service would need a means of representing digital-imaging goals and their levels.
- The service would need a means of mapping goals down to measurable events.
- The service would need a theory of "treatment" that linked tool configuration with skills and goals.

Frankly, we think all the above are obtainable. In fact, we are discussing among ourselves an open source project that might produce a digital imaging tool that could act as the base for monitoring and adaptation, which in turn is linked to goals and skills. If we are successful, the problem becomes one of economics. How much would such a service need to charge to stay in business, and how much would people like Fickas be willing to pay? This, we believe, returns us to the topic of section 5: scale and automation. Perhaps the assessment process could be automated through computer-based testing. Perhaps a user could self-choose and self-refine their goals from known goal patterns. Perhaps monitoring and adaptation could happen automatically, delivering the user a tool that is just right for their goals and skills at any point in time. With this level of automation, we believe personal software services that mimic clinical services have a future: users who share the same personal goals can be taken at their own pace through a set of tool configurations. What if there is less than full automation, e.g., expensive staff must be in the loop? What if goals are totally idiosyncratic, e.g., no one shares Fickas' goals? What will markets-of-one bear in terms of cost to pay for such a service? Our one data point, Fickas, is happy to say he would pay something, but is much less certain how much.

8 Acknowledgments

The National Institute on Disability and Rehabilitation Research has been a generous contributor to the work reported here. This material is based upon work supported by the National Science Foundation under Grant No. 0234571. As such, any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

This paper reports on a team effort. We acknowledge the contribution of the hard working students on the team over the last four years: Jim Allen, Moira Burke, Andrew Fortier, Harrison Green-Fishback, Jason Prideaux.

References

1. BIA. BIA Community Awareness Presentation, Brain Injury Association, Inc, 2000.
2. Dardenne, A., van Lamsweerde, A. and Fickas, S. Goal-Directed Requirements Acquisition. Science of Computing Programming, 20. 3-50.
3. Dwyer, M.B., Avrunin, G.S. and Corbett, J.C., Patterns in property specifications for finite-state verification. in Twenty-First International Conference on Software Engineering, pages, (Los Angeles, 1999), 411-420.
4. Feather, M.S., Fickas, S., van Lamsweerde, A. and Ponsard, C., Reconciling System Requirements and Runtime Behavior. in Proceedings of the International Workshop on Software Specification and Design (IWSSD'98), (Isobe, 1998), IEEE CS Press.
5. Fickas, S. and Feather, M.S., Requirements Monitoring in Dynamic Environments. in Proceedings of the 2nd International Symposium on Requirements Engineering, (York, England, 1995), IEEE Computer Society Press, 140-147.
6. G. Fischer, "User Modeling in Human-Computer Interaction," User Modeling and User-Adapted Interaction, vol. 11, 65-86, 2001.
7. Friedman-Hill, E. Jess in Action. Manning Publications Co, 2003.
8. Hui, B., Liaskos, S. and Mylopoulos, J., Requirements Analysis for Customizable Software: A Goals-Skills-Preferences Framework. in The 11th IEEE International Conference on Requirements Engineering, (Monterey Bay, CA, 2004), 117-126.
9. LoPresti, E.F., Mihailidis, A. and Kirsch, N. Assistive technology for cognitive rehabilitation: State of the art. Neuropsychological Rehabilitation, in-press.
10. D.A. Norman, Emotional Design: Why We Love (or Hate) Everyday Things, New York: Basic Books, 2004.
11. C. Potts, "ScenIC: A Strategy for Inquiry-Driven Requirements Determination," 4th IEEE International Symposium on Requirements Engineering, 1999, 58-65, Los Alamitos CA: IEEE Computer Society Press.
12. J. Robertson and S. Robertson, Mastering the Requirements Process, Harlow: Addison Wesley, 1999.
13. Robinson, W., N. and Pawlowski, S., D. Managing requirements inconsistency with development goal monitors IEEE Transactions on Software Engineering, 1999, 816-835.
14. Robinson, W.N., Implementing Rule-based Monitors within a Framework for Continuous Requirements Monitoring, best paper nominee. in Hawaii International Conference On System Sciences (HICSS'05), (Big Island, Hawaii, USA, 2005), IEEE.
15. Robinson, W.N. Monitoring Requirements Development with a Goal Model, in. in Jeusfeld, M., Jarke, M. and Mylopoulos, J. eds. The Method Engineering Textbook, MIT Press, 2004.
16. Sohlberg, M.M., Ehlhardt, L.A., Fickas, S. and Sutcliffe, A. A pilot study exploring electronic mail in users with acquired cognitive-linguistic impairments. Brain Injury, 17 (7). 609-629.
17. Sohlberg, M.M., Fickas, S., Ehlhardt, L. and Todis, B. Case Study Report: The Longitudinal Effects of Accessible Email for Four Participants with Severe Cognitive Impairments. Journal of Aphasiology, in press.
18. Sohlberg, M.M., Fickas, S., Hung, P. and Lemoncello, R., Community Navigation Profiles for Six Individuals with Severe Cognitive Impairments, Poster session. in The annual meeting of the National Academy of Neuropsychology, (Seattle, WA., 2004).

19. A.G. Sutcliffe, User-Centered Requirements Engineering, London: Springer-Verlag, 2002.