

Preferences for Global Access:
Profile Creation Support for Cloud-Based Accessibility

Institute for the Study of Knowledge Management in Education (ISKME)

Design Report

October 23, 2013

Prepared by:
ISKME
Ontario College of Art and Design University
WGBH
IBM
Raising the Floor International
Inclusive Technologies

Inquiries about this work can be addressed to lisa@iskme.org

ISKME: 

This report is part of the Preferences for Global Access (PGA) project. The PGA project was funded by the National Institute on Disability and Rehabilitation Research (NIDRR), a component of the U.S. Department of Education's Office of Special Education and Rehabilitative Service, under contract ED-OSE-12-D-0013 for Profile Creation Support for Cloud-Based Accessibility. Inquiries about this work can be addressed to Lisa McLaughlin at lisam@iskme.org.

Suggested Citation: Petrides, L., McLaughlin, L., Jimes, C., Brennan, M., Treviranus, J., Schwerdtfeger, R., Rothberg, M., Vanderheiden, G., Tobias, J., Trewin, S., Clark, C., Mitchell, J., (2013). *Preferences for Global Access Design Report: Profile Creation Support for Cloud-based Accessibility*. Unpublished manuscript, submitted to the U.S. Department of Education's Office of Special Education and Rehabilitative Service.

About this Report	4
1. Background and Need	5
2. PGA Project Objectives.....	6
3. Complicating Factors	7
3.1 Diversity of the Users	7
3.2 Cognitive Access	7
3.3 Personalization.....	8
3.4 Solution Readiness and Awareness	8
4. Project Roadmap	9
5. Progress in Year One	10
5.1 Process	10
5.2 Results.....	10
5.2.1 Preferences Matrix and Cognitive Preferences	11
5.2.2 Guiding Concepts	12
5.2.2.1 Conceptualized User Activity Spaces.....	12
5.2.2.2 Identified Interaction Methods	13
5.2.2.3 Identified Inference Best Practices.....	15
5.2.2.4 Identified Core Design Requirements.....	17
5.2.3 The Preference Tool Ecosystem	18
5.2.3.1 Identified Categories of Tools in the Preference Tool Ecosystem.....	18
5.2.3.2 Defined Specific Roles and Design Constraints for the Preference Tools.....	20
5.2.3.3 Defined Ways that Inference Can Be Used Within the Preference Tools.....	24
5.2.4 The Preferences Framework: Architecture Design	26
5.2.4.1 Identified Key Architecture Design Constraints.....	26
5.2.4.2 Identified Preference Framework Components	33
5.2.4.3 Conceptualized Developer Tools	37
6. Next Steps in the PGA Roadmap.....	40
6.1 Gather End User Input on the PGA Ecosystem and Framework	40
6.2 Develop Preference Tool Ecosystem and Preferences Framework Exemplars	40
6.3 Enable a Community of Designers and Developers.....	41
6.4 Develop an Extensible Knowledge Store to Guide Outreach	41
7. Conclusion	42
Appendix A: Use Cases.....	44
Appendix B: Dissemination of the PGA Work.....	45
Appendix C: Project Background Information	49
References	50

About this Report

This report is the Preferences for Global Access (PGA) design deliverable for contract ED-OSE-12-D-0013, Task Order 0001. The goal of this work is to provide essential tools for preference discovery and management as well as to encourage ongoing development of preference editing tools within the community at large. This effort informs the work of building the Global Public Inclusive Infrastructure (GPII).

This report describes tools and a technical architecture to support preference discovery and management, and begins to describe how a system developed from these foundations can be sustainable and integrated with existing work on user needs and preferences sets.

The report is divided into 7 sections.

- **Section 1** presents the **background** to the project, and highlights key needs surrounding preference discovery and management;
- **Section 2** outlines the **objectives** of the PGA project in light of these needs;
- **Section 3** introduces the **complicating factors** that must be addressed in reaching our project objectives;
- **Section 4** presents a **roadmap** of this work for year one and beyond;
- **Section 5** presents the **progress** made in year one, our approach and the results of our work;
- **Section 6** covers the recommended work to be done in the **next phase** of the effort —including the creation and deployment of a functional prototype for the tools and architecture, engaging user feedback, identifying what works for specific user requirements, and creating scalable tools; and
- **Section 7** provides the **conclusion** to the report.

The report includes three appendices.

1. **Appendix A** presents design work to visualize the landscape of work (preference discovery, declaration, and management), several example use cases that explore why these tools are needed and how they will be used, and exemplar prototypes that show how specific individuals might use these tools;
2. **Appendix B** discusses the dissemination efforts of the PGA work by project team members; and
3. **Appendix C** provides an overview of the relationship between the PGA work and GPII.

1. Background and Need

Each year, our daily activities are increasingly mediated in some way by digital technologies—from point-of-sale devices, to self-service kiosks, to Internet-based services, to name a few. Access to these technologies has gone from optional to essential as more traditional ways of doing things are replaced with devices having digital interfaces: Agents are becoming kiosks, books are becoming eBooks, and parking meters are becoming digital fare stations. As such, the ability to use digital technologies is all but required for full participation in commercial, cultural, recreational and social activities. Yet today, millions of people are not able to access these technologies.

While there have been advances in accessibility features embedded in native platforms and websites, and while advocacy toward accessible user interfaces and content design continues, most applications and products are designed as one-size-fits-all solutions. For example, if users are blind pictures are replaced with text, and if users do not use a mouse, keyboard navigation is activated. This approach fails to recognize the heterogeneity of individuals and the different contexts they operate. Access issues arise for multiple reasons, including breakdowns between the technology's interface and a user's ability to engage with it, and compatibility of access solutions with a user's device or with a user's context or setting. Additionally a user's functional disability may be temporary or permanent, stable or progressive. This means that a user may have shifting needs over time that move within or between certain diagnoses. In sum, accessibility solutions designed for people with a specific disability diagnosis are not fully inclusive, and only provide support for individuals whose problems can be described as "typical" for that diagnosis. The result has been a limited set of solutions that work for some, but exclude many.

The predominant one-size-fits-all approaches not only affect the "goodness of fit" and thereby effectiveness of the solutions, but also remove the ability of individuals to control the interfaces they must use. Whether embedded in the design of products or services, or embodied in the clinical interventions that arrogate such decisions to certified authorities, current solutions potentially deny agency—the sense of autonomy made concrete by action—to users. In doing so, these solutions can serve to disempower the person who is theoretically best positioned to make his or her own personalization decisions and to learn from those decisions.

The result of factors such as these is that many users are dissatisfied with their assistive technology (AT) systems (Mann et al., 1993; Hastings Kraskowsky & Finlayson, 2001; Fisk et al., 2009, Lenker et al., 2013), and many report underutilizing or abandoning their systems altogether (Reimer-Weiss & Wacker, 2000; Scherer et al., 2005). In attempting to mitigate these factors, this project seeks to replace a one-size-fits-all, "inoculation model" of accessibility—one assessment and prescription for all time—with an approach that is more accurately developmental and attuned to technological churn. We seek to enable users with sensory or physical limitations, cognitive constraints, unique learning affordances, or other barriers to discover and articulate their needs and preferences for specific contexts, in an unobtrusive yet helpful way. Enabling users to explore preferences and to learn how to improve their experience

with technology can have a significant impact on users' ability to remove the barriers to access that they experience.

2. PGA Project Objectives

Within the scope of this project, preferences are defined as functional descriptions of how users prefer to have information presented, how they wish to control any function in the technology application, and settings or commands that are stored (or that will be stored) for them. Preference sets are groups of preferences, which may apply to different contexts or devices. One user may have multiple preference sets, such as preferences for home, for school, for their desktop, or for their mobile device.

This project is focused on creating, refining and maintaining preference sets for users in order to support **access to information communication technology (ICT)**, including technologies, tools, devices and resources; **ease of use of ICT** by all users, including non-technical users; and **increased participation** by individuals in all of the activities that require ICT or that ICT can enable, including the activities that establish their needs and preferences.

The specific objectives of this project (Year One and beyond) are to:

1. Provide a knowledge base on proposed user preference gathering systems for people with disabilities, including expert judgment on how best to assist users in specifying the appropriate aspects of these preferences.
2. Develop a system of tools and technological architecture that supports creation of user preferences for people with disabilities, is operable as a Web application or system of applications that reflects expert judgment, and supports continuing evolution of the content of user preferences and of the means of assistance provided to users. These tools should:
 - Allow users to create initial preference sets and preference sets for different contexts;
 - Encourage users to manage their preference set(s) through direct manipulation/editing, and by responding to suggestions created by inference engines monitoring their use and contexts—the aim being to support users over time, as their needs and preferences change, and as they develop increased understanding of what they require and what works best for them through the use of the tools;
 - Engage users, reduce the time it takes them to acquire their preferences, and allow them to quickly make changes to those preferences throughout the lifecycle of interacting with ICT; and
 - Be appropriate to different user groups—from power users to those who have limited experience with technology; to users who may not know what they want and need; to users of different ages, in different environments, and who utilize different technologies.
3. Aggregate and refine knowledge regarding what works for specific needs and preferences from a variety of knowledge sources including professionals, user feedback, and usage metrics, so that this can be used to guide the user in selecting preferences

4. Encourage ongoing development of preference tools within the community at large

Before turning to the PGA roadmap for reaching the above objectives, the following section raises challenges, or complicating factors, that we deem as central to address in our work.

3. Complicating Factors

In line with the challenges outlined in Section 1, there are a number of complicating factors that make the creation of preference tools particularly difficult to do well.

3.1 Diversity of the Users

In considering all of the different types and degrees of disability that an individual may have, it is apparent that complex patterns of user needs occur even within a single disability type. The needs of someone who is completely blind, for example, are different from someone who has low vision and can see but cannot read text. And the needs of the individual who cannot read text are different from an individual who requires screen magnification for readability. Furthermore, individuals can and often do have one or more disability concurrently—including a combination of both physical and cognitive disabilities. At the same time, individuals vary in terms of their technological self-efficacy, and some may not use computers at all. Given these complexities, creating tools that will work for the diversity of individuals is a challenge. For example, how can we create a computer-based tool to help novice users set up their first preference sets if they cannot perceive, operate, or use the computer? Accessibility solutions must be flexible and responsive enough to balance the need to overcome serious barriers to entry with the need to showcase diverse preference possibilities to users who are interested in them. Ultimately, accessibility solutions must meet the unique needs of each user.

3.2 Cognitive Access

Cognitive needs potentially impact the widest number of users; however, accessibility standards and solutions have not done an adequate job of addressing this group of users (Hudson et al., 2004). This is due to several factors, including the complexity and wide variation in the realm of cognitive needs, that there is a lack of developer education on the needs of cognitively impaired users, and that there is a lack of a single vehicle for capturing cognitive needs. When solutions do exist, they are either specialized stand-alone solutions with limited scope or are limited to a single application or web browser (e.g., the MathPlayer plug-in which is limited to run in Internet Explorer). In some cases, a limited focus on necessary requirements to specific cognitive constraints ignore their incongruence—and in some cases incompatibility—with preference solutions for other disabilities.

There is also an inherent paradox: some of the solutions intended to support individuals with cognitive needs, such as the timing and modality of reminder messages, may be beyond the cognitive ability of the user to comprehend and accurately select these preferences. This raises the potential role of

assistance or helpers in preference collection. It is inherently clear from the research that without preference gathering tools capable of acquiring the varying needs of users we will not be able to adequately help users facing cognitive access barriers.

3.3 Personalization

Research has shown that when users are provided with the opportunity to change their preference settings, they rarely do so (Iyengar et al., 2000; Trewin, 2000; Forrester, 2004; Spool, 2011). Trewin (2000) discusses barriers to personalization of preferences for less experienced users in particular, including lack of awareness of available options, lack of knowledge of how to change settings, difficulty identifying the solution to meet a given preference or need, and a lack of control over the unconfigured interface. This presents a challenge, especially in light of the fact that assistive technology (AT) solutions do not provide “session support”—a way for users to easily explore and get assistance if needed when creating or changing settings. A recent study by the Pew Research Center found that 63 percent of non-Internet users would need assistance in getting started on the Internet (Zickurh, 2013). And an earlier study by Pew identified people with disabilities as 27 percentage points less likely to use the Internet than their non-disabled peers (Fox, 2011).

Thus, it is somewhat unrealistic—based on these and other, similar research findings—to expect large numbers of people who do not use the Internet, and who are concerned that they may not be able to use it, to do so with independence and confidence. However, the ability to personalize does not negate the need for or importance of adaptation for these individuals: The fact that it is difficult for users to understand or configure technology to meet their needs should not be confused with the critical importance of being able to do so. Bridging this gap is one of the key goals of this project.

3.4 Solution Readiness and Awareness

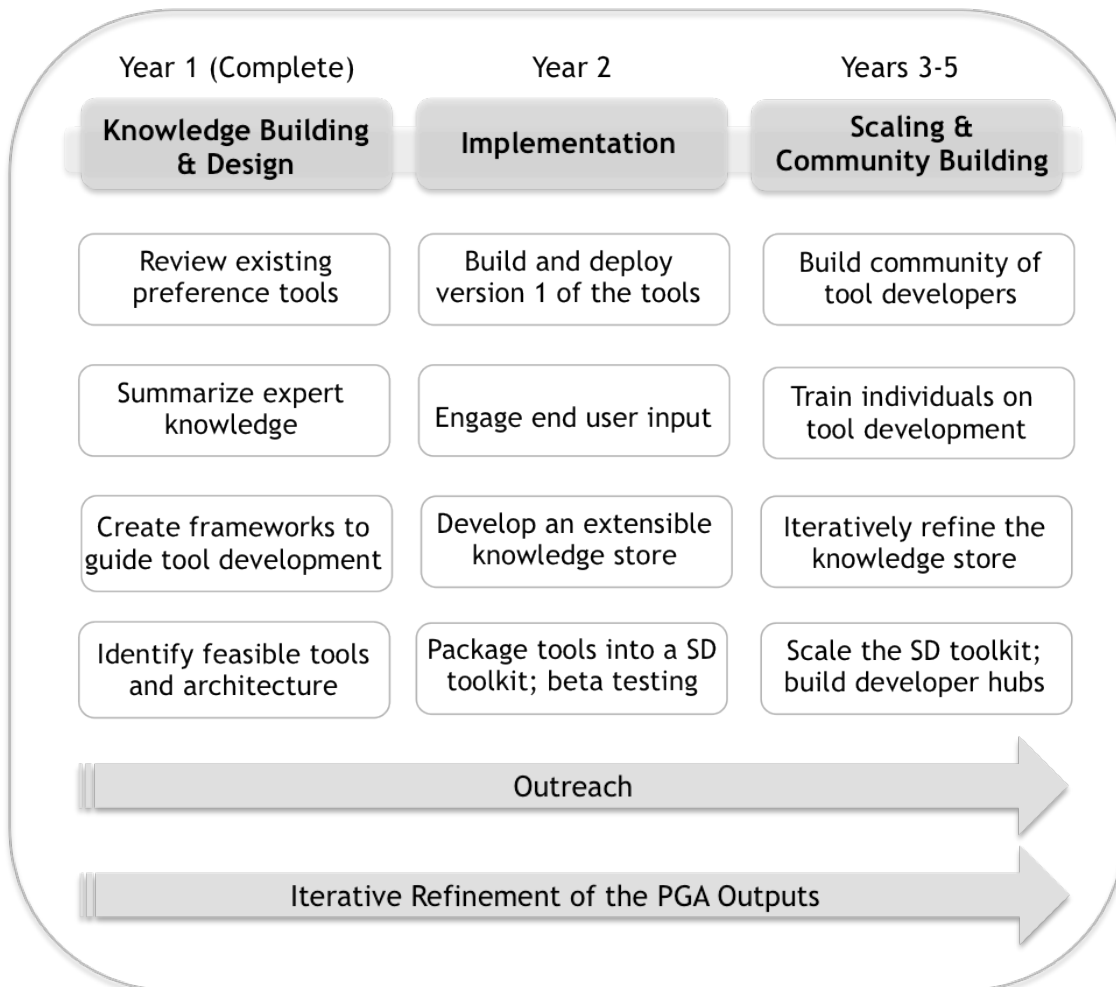
Beyond the complicating factors listed above tied to user-centered design, this project seeks to address the complexities involved in creating scalable software architecture. The creation of globally accessible and highly personalizable solutions require that the software is designed to be easily understood, extended, and adapted for use across many platforms and contexts. One-size-fits-one solutions, which this project seeks to build upon and create, are in the early stages of readiness for mainstream consumption. The gap between potential solutions identified by cutting edge research and readiness to implement them is currently wide. We foresee that in some cases, challenges may arise due to unwillingness on the part of proprietary platform providers to adopt open source solutions, in other cases, developers may not have a clear understanding of the concepts or systems used to build and implement tools. Furthermore, some who are interested in improved solutions may not have the financial resources to devote to develop them. Widespread adoption of the models presented here within and beyond the accessibility community requires further work towards a standardized framework of development concepts and practices, and well-defined pathways for contribution and adaptation. As

such, we must address interoperability and platform dependency challenges, find ways to support non-experts in understanding and integrating tools into their software systems in a scalable way, and recognize and attempt to address the cost and time intensive implementation processes.

4. Project Roadmap

In an effort to mitigate the above challenges through the creation of a preference architecture and exemplar tools and systems, the PGA project team has constructed a roadmap depicting the full scope of how this work would look, including and beyond Year One. The roadmap is depicted in Figure 1, below:

Figure 1. Proposed PGA Project Roadmap – Year 1 and Beyond



As shown above, Year One has focused on building knowledge and a set of frameworks to inform the identification of feasible preference tools and architecture. Year Two will focus on implementation of the first iteration of these tools, as well as user testing and packaging the tools into software

development (SD) toolkits, to support the use of the tools in future contexts and by future users. The remaining three years of the PGA work will focus on scaling and building community around the tools. The work of Year One is detailed in Section 5 below, and plans for Year Two will be discussed at the close of this report (Section 6).

5. Progress in Year One

5.1 Process

Year One of the PGA work consisted of a series of research efforts, PGA project meetings and listserv conversations, and external outreach conversations and presentations. The research efforts, which resulted in [deliverable 1](#), included a review of literature to capture extant theory and empirical work in the preferences space, and a review of existing user needs and preferences set systems and accessibility initiatives. The subject matter experts (SMEs), as part of the PGA project team, utilized the findings of the research to extend and support their understanding and experience in the field. Mechanisms were also instilled to solicit feedback and input from other field experts, outside the project group, and to share project insights with individuals working with or relevant to preference discovery and management (see Appendix B for a list of outreach efforts).

Through this process, the PGA project team made considerable progress in better understanding the preference space, and in identifying subtle and not-so-subtle problems in creating practical tools. Together we began to articulate the most effective ways to approach defining and presenting preference specifications to address these problems. The insights from these activities were shared and discussed in the project's weekly design team meetings and through the PGA listserv—culminating in the recognition that we are at a key technological transition point that moves away from supporting users through aggregated functions that fit their needs poorly, toward supporting users in assembling personalized interfaces based on needs and desired functional tasks.

5.2 Results

As a result of Year One, the PGA team delivered three critical outputs in the field of accessibility that will have a transformational impact on this project and the industry as a whole: 1) We deepened understanding of the diversity of user preferences, including cognitive access and advocacy momentum to guide new forms of support for users with cognitive impairments; 2) we created the Preference Tool Ecosystem, which is a set of concepts and paradigm-shifting maps for software development toolkits for preference acquisition; and 3) we designed specifications for the Preference Framework—an architectural framework capable of supporting exchange of preference tools and data across platforms and networks.

Building on our knowledge inventory around the diversity of user preferences, the central output of our work has been a map for the creation of flexible software development kits to support users in discovering and saving preferences entitled the Preference Tool Ecosystem. The team created a flexible model through the project that places users at the center and employs novel methods for engaging users in a preference discovery process. These approaches include methods that aid users in learning how they learn best, a strong inventory of best practices for inference driven approaches to preference acquisition aiming to make the learning to learn process more intuitive, and conceptual frameworks that outline ways of thinking about interface methods and user activity spaces that can be utilized to understand the ecosystem and guide tool development. Finally, we describe the three general categories of tools that inhabit the Preference Tool Ecosystem: initialization tools, exploration tools, and editors.

Based on our growing understanding of the Preference Tool Ecosystem we developed deepened knowledge of architectural specifications that would be needed to support interoperability between the kinds of tools that would inhabit the ecosystem. The PGA architecture section outlines the Preference Framework, which effectively ties new design thinking for preference acquisition generated by our roadmap to existing GPII architecture while at the same time defining extensions to the GPII architecture to fill gaps needed to support preference inference. New browser and platform limitations and challenges have been identified and catalogued here, new specifications articulated, and a wealth of opportunity unearthed for the establishment of a community of developers to build out and scale the solutions identified here.

The following sections describe our results in greater detail.

5.2.1 Preferences Matrix and Cognitive Preferences

The [preferences matrix](#), developed in Year One as part of our knowledge building work, contains an extended list of disabilities and needs that moves beyond existing work in the accessibility and special education literature, to include cognitive needs, contextual needs, and individual learning styles with examples of their functional equivalents, or preferences.

In compiling the preferences matrix our work revealed that although cognitive needs, in particular, potentially impact the widest number of users, accessibility standards and solutions have not done an adequate job of addressing this group of users. The PGA team produced a concise inventory of requirements needed to support users with cognitive impairments that revealed opportunity spaces for advancing the field such as awareness of the deep inadequacy of current digital math solutions capable of serving users with cognitive constraints, insights around the potential impact of widespread use of well-known technological supports like breadcrumbs on access for users with cognitive constraints, and guidance for where to get started in an area long-regarded as too difficult to design for.

Subject Matter Experts from the PGA team are taking an active role in translating PGA project findings on cognitive needs into the development of a cognitive roadmap by initiating the formation of W3C Cognitive Accessibility Task Force. The task force will work directly with the Web Accessibility Initiative Coordination Group to execute on the strategy designed by the Task Force. In return the other aspects of the Cognitive Accessibility Task Force will help to inform and guide efforts to build a roadmap in the World Wide Web Consortium to address auto-personalization of preferences for cognitive access.

5.2.2 Guiding Concepts

In order to successfully build the tools to support preference gathering in line with the PGA project goals and challenges, it was critical to develop ways for understanding how users enter into and move around in preference environments, how to efficiently capture preferences, and which interaction methods and design strategies are most suitable to engage and support users. Thus, to lay the groundwork for the Preference Tool Ecosystem, the team created concepts and conceptualizations around user activity spaces, interaction methods, inference methods, and tool design. Each is detailed below.

5.2.2.1 Conceptualized User Activity Spaces

Considering the diversity of users, it became clear in our work that users would require different points of entry into preference collection activities. They would need to choose how, where, and when they engage with the tools, all in light of their context, diverse experiences, attitudes, and access to support. From this foundation of user and contextual diversity, the team identified the activities users would potentially require for meaningful engagement with preferences, and created names for various “activity spaces” users would inhabit.

These activity spaces were identified and documented to ensure that our design approach keeps the users central, and that the tools developed meet users where they are. They were also created to emphasize that different tools will be needed in order for the full range of users to interact with at least one tool that matches their needs—regardless of their expertise, abilities, or any other factor that might reduce participation. The five activity spaces developed by the PGA team include:

1. **First Create** - This activity space is essential for users who are not already computer users. When in this activity space, the novice user is supported in determining what is needed in order to be able to use the computer (or other devices) at all. (If users can already use the computer, then they could use CAPTURE to start with the settings they already use. From there, they could go into an EXPLORE activity to find out what really works for them. If the user has no useful ability to operate the computer – then First Create would be used. After First Create, that user would likely then use Explore.)

2. **Capture** – If users are already using a computer, their current settings can be used as a starting point, understanding that default settings should not be presumed to serve the user well. Capture can also be used at any time to capture a particular set of settings – for example to capture a new set to be used for a particular context such as a set of preferences that work well for that user in libraries.
3. **Explore** - In the Explore activity space, users who have at least basic access, are encouraged to try new configurations, find new transformations, and discover new preferences. A successful exploration will allow users to explore adaptations without fear of permanently losing their current level of access because they want to try something new to see if it is better.
4. **Adjust** - This activity involves providing the user with the ability to easily adjust a single preference, or adjust multiple preferences. This also allows users to: a) adjust the setting of a preference (i.e. change the size of the text for a text enlargement preference); b) undo previous settings; and c) instantly apply changes to content that the user is viewing (independent of whether or not the user saves or stores).
5. **Manage** - Manage covers a wide range of advanced activities involving the viewing, editing, and/or re-organizing of preference sets. In a tool that enables “manage”, a user may choose to review all of the different preference sets; add or remove preference sets for specific contexts and devices (i.e. home or school and phone or desktop); create new preference sets for different contexts and devices; share and/or sync preference sets with other users; and provide feedback to the tool. The manage activity space encompasses advanced features that many users may never require or may prefer to not use.

It is important to note that within this conceptualization, users will not necessarily move through the five activity spaces in a linear progression, but rather the activity spaces represent different ways that users may interact with preference tools in different contexts. Furthermore, the activity spaces do not provide an indication of how the user will *perform* the various activities. To actualize these five user activity spaces, interfaces must be designed using various methods, discussed below, that enable users to make preference selections. Combining different interface methods with thoughtful visual and interaction design begins the work of tool design.

5.2.2.2 Identified Interaction Methods

Interaction methods are the techniques for presenting and providing preference features to users. The aim is to offer appropriate methods within each activity space to engage the user to discover, explore, and adjust preferences—toward the ultimate goal of increasing user engagement and success in completing the preference process. The methods may be determined by the designer of a particular tool (e.g., there may be many different tools that support different audiences of users in exploring preferences), and/or selected by users, as they move within a preference environment that has multiple tools.

Based on a review of the literature on what works for a given type of interaction, and through PGA team discussions of exemplar preference tools, the PGA team identified five categories of interaction methods. Listed below. These interaction methods are not mutually exclusive and will likely be combined in various ways to create flexible interfaces that meet users where they are, engaging them in a way they feel comfortable.

- **Prompt & Ask** - The most direct method is to explicitly prompt the user to make a selection for a preference. One technique is to ask a direct question such as “Do you want captions to be shown when you watch a video?” This works well when users know their own needs and preferences, or when the options are few or can be clearly presented (e.g. fonts of different sizes). A person with a visual impairment might know that they need 300% screen magnification, without needing to use a slider or other indirect method of selection. However, some users may need the opportunity to try an option, or to compare two options and choose between them. This works well when users do not know in advance what they need or want, and when the preference is something the user is just discovering for the first time.

Guides, tips, and questions may supplement a “prompt & ask” method and include: in-context guides when new features are introduced, tooltips for various controls, or questions a user must answer in order to determine the next step in the process.

- **Tasks & Rewards** - These may include game-like interactions that are highly engaging and stimulating, or more straightforward questions or a combination of both. These methods would present a clear goal and provide a reward or encouragement, while collecting preference data more indirectly than other methods. One example might be a game that involves clicking on targets. The user’s game performance provides data from which the system can infer appropriate control sizes and spacing. Another example is a progress bar indicating how complete someone’s preference set or profile is. Tasks can be combined with guides to make powerful and engaging tutorials.
- **Show** - The user may wish to have the system show specific preferences or types of preferences that are available, as well as the specific adjustments that are available for each preference. This may be useful for users who know what they are looking for and want to set specific preferences quickly.

One form of showing is to present advanced search menus of preferences. Preferences can be filtered in many ways. Users can select a curated category such as “Speak Text” or “Text Alternatives” as a starting place to view related functional supports. Users can also search for preferences using natural language or technical/medical terms (e.g., “tired,” “text size,” “asthenopia”). Users can also simply view the full preference list in alphabetical order.

Users can also customize preferences individually by setting specific values such as setting the foreground color to white and background color to black. Or a user can select a preset or group of preference values as a quick starting place for more granular adjustments, thus enabling multiple adjustments in a single-action, for example, a preset of “Easier to see” could change foreground and background colors, set the font to sans-serif, and activate the highlighting of links & buttons. Presets make it possible to introduce users who are not familiar with specific settings to a variety of preferences they could benefit from, while preference adjusters allow users to get more specific about their needs. The choice of preferences to include in presets can be informed by prior inferences based on user performance. The concept of presets is a rich one and one that deserves further work.

- **Randomize & Try** - When not much is known about the user, a preset or common selection of customizations (or selections predicted from other information gathered about the user) could be presented for the user to preview. They could try the customizations offered or ask for a new set of choices. For example instead of 3 controls to adjust font, size, and spacing – there might be several examples and they click one to set all three.
- **Infer & Recommend** - With so many possible preferences, and possible values for each preference, and possible interactions between preferences, users cannot try everything. They may become overwhelmed by the preference selection process itself, and abandon it. In order to make the process more successful for more potential users, it will have to be as simple, pleasant, and short as possible. This is where inference can be a valuable method. Many techniques can be used to infer preferences that are likely to be relevant to the current user, and these preferences can be prioritized or highlighted in the user interaction (e.g. using “prompt and ask” or “tasks and rewards” methods). With good inference, a user may simply need to try a few suggested preferences and decide whether to add them to their profile, instead of being faced with a long list of preferences that are mostly irrelevant to their needs. The role of inference in preference tools is discussed in more detail in Section 5.2.2.3, below.

These five interaction methods can be used together within the interface, can be presented individually, and can be opted into through user choice. Once a marketplace of PGA tools exists, developers and designers of tools will be able to make use of plug and play versions of tools, or build a tool from scratch that employs interaction methods that best suit the users they hope to reach. It is recommended that potential implementers familiarize themselves with the methods identified here as a starting point.

5.2.2.3 Identified Inference Best Practices

As discussed briefly in the section above, inference is a technique for suggesting appropriate preferences that is not based on explicit user direction. Most of the research on the inference of accessibility settings has focused on analysis of user actions, with the goal of recognizing when users are

having difficulty, and calculating appropriate settings to improve their accuracy or efficiency. Inference methods have been successfully demonstrated for several keyboard, (Trewin, 2004) mouse (Gajos, Wobbrock & Weld, 2007), and switch input (Leshner et al., 2000) parameters, and early indications are that touchscreen configuration preferences would also be feasible (Nicolau & Jorge, 2012). The most successful use of inference is likely the case where the means of inference is based on physical user actions.

Inference-based approaches are not suitable for all preferences. To explore what kinds of preference are amenable to inference-based methods, the PGA project team considered a large set of possible preferences, generated by the project (see the [preferences matrix](#)), and gathered information on the state of the art and likely future progress. The following, tentative conclusions are thus based on the team's experience, and known prior work:

- Preferences that control aspects of the way input devices respond are often measurable, both with and without knowledge of the user's specific task. Examples include key repeat delay, key repeat rate, debounce time, acceptance delay, scanning rate, one-digit interaction, touch time, mouse speed, double click speed. It is especially useful to measure ideal values for these preferences, as this can avoid a trial-and-error cycle of finding the best setting.
- Sometimes it is better and simpler to ask. Examples include the user's preferred text input method (e.g., on-screen keyboard or physical keyboard), and desire for alternative forms of content.
- When a user is unaware of a particular assistance technique, for example video captions, their preference for that technique cannot be inferred. Users will need to be shown an example in order to decide their preference. It may, however, be possible to infer that the technique may be of interest, based on comparison with similar users, or the user's other preferences.
- Using eye-tracking data could enable new inference techniques, especially techniques to measure the impact of different visual features such as cursor size, mouse trails, font size, color contrast and spacing of letters, words, lines and paragraphs. Eye tracking laptops and glasses are already on the market, and many people are using mobile devices with built-in user-facing cameras. Without eye tracking, the effectiveness of different visual presentations could be estimated using tasks that involve reading and then pressing a button. The more quickly the passage is read, the better the readability. However, this does not allow us to identify an ideal setting – only to evaluate options presented to the user, which the user can often do themselves.
- A few preferences are easier to observe in normal system use than to measure in a test setting. For example, the user's preferred navigation method within a web page (e.g. tab key, headings, direct touch access) can be observed, or their use of dictionary and spell checking tools, or keyboard shortcuts.
- User behavior within a tool could be used to flag some preferences even when the user's intended action is not known for sure. The flag may then trigger a more structured measurement interaction. For example, if a user clicks outside a target, preferences for larger targets and reduced mouse sensitivity could be flagged for exploration by the user later in the session.

In addition to the above, it is important to note that needs and preferences that have been inferred are not always accurate. Even if they are accurate, users may not want them included in their preference profile. For example, users may be using a device with one hand because they are holding an object in their other hand. The ideal role of inference in preference gathering is to provide *suggested* preference settings. These can be used to inform and drive user interaction to make preference selection more efficient. The final decision on whether to add any given preference to a preference profile must rest with the user, or his or her helper or representative.

Implementations of inference should also avoid prescriptive language (telling the user what he or she needs) when making suggestions. User autonomy and engagement can be supported by encouraging users to try the suggestions and form their own opinions. This does presume that users are able to make preference decisions. Decision-making is easier when users can experience the proposed setting, rather than reading a description. For example, users should be able to try out a suggested scanning delay or target size before making a decision.

In looking at the use of inference and preferences it is important for users to always have control of any changes to stored preferences. Although inference techniques can be used to support real-time automatic adaptation of systems, using inference to change a user's preference set is a very different use case because the preference set is intended to be more or less permanent, comprehensive, and is often globally applied. Users need the ability to know and control what information about them is stored in user preference sets. When an algorithm infers a potential preference value, such as a scanning delay of 1.2 seconds per item, user confirmation is needed before adding this to a user's preference set. Given this inference, a preference-gathering tool can suggest a 1.2s scanning delay to the user, and/or ask the user to try it and make a decision ("Prompt & Ask"). The final decision whether to adopt this scanning delay rests with the user, or their authorized agent.

Finally, inference methods are not available for all types of preferences. We are not aware of existing techniques for measuring a user's need for learning aids such as graphic organizers, graphing calculators, and progress trackers.

5.2.2.4 Identified Core Design Requirements

We identified nine concrete requirements for preference tools if they are to meet users where they are, rather than expecting them to adapt to yet another system with unfamiliar interactions. The preference tools should cumulatively:

1. **Encourage exploration and experimentation** by users, through mechanisms that support them in safely trying out new preferences. Tools should accommodate the unexpected and allow users to express new preferences that may arise for them, and to discover potentially unanticipated and unexpected preferences they may have.

2. **Demonstrate the chosen preference** wherever possible through realistic simulations of specific preference choices, or instant implementation of preferences. This helps the user understand the function of the adjustment that they have chosen.
3. **Ensure that users are informed** of anything that is being done to their preferences, as a means of making the preference management system transparent.
4. **Ensure that users are allowed to control preference selections.** Automatic adjustments should only be with user consent and control.
5. **Integrate preference management into user workflow**, by, for example, offering users the option to make just-in-time decisions by monitoring and measuring user actions and providing automated suggestions for preferences based on those actions.
6. **Enable continuous refinement**, so that users can choose context-specific or session-specific preferences, as well as more generic or global preferences. The user should also be able to undo previous sets or settings.
7. **Integrate usable decision support wisdom**, by providing feedback to the user on what preferences have worked in similar circumstances for other users, or by enabling users to communicate directly with peers, experts, and leaders.
8. **Provide a range of interfaces, ranging from ultra-simple to full-featured** – to enable preference control by users at all skill levels.
9. **Where possible, infer and suggest user needs by monitoring** their interaction with the system to improve the user experience.

5.2.3 The Preference Tool Ecosystem

5.2.3.1 Identified Categories of Tools in the Preference Tool Ecosystem

As our understanding of users, activity spaces, and interaction methods matured we were better able to evaluate our existing set of preference tools and to develop new tools that fill identified gaps. This helped us to understand how these tools could work together systematically. We defined this system as the Preference Tool Ecosystem.

The Preference Tool Ecosystem is composed of three general categories of tools: initialization tools, exploration tools, and editors. Within each of these categories, we have identified specific tools aligned to each of the user activity spaces outlined earlier in this report (First Create, Capture, Explore, Adjust, and Manage), and these have been adopted by the overall GPII ecosystem (see [http://wiki.gpii.net/index.php/PCPs, PMTs, etc](http://wiki.gpii.net/index.php/PCPs,_PMTs,_etc)). The categories of tools, and specific tools that fall within each, are as follows:

- **Initialization Tools** provide *First-Create* or *Capture-Create* functions; they help users to get into the door, and to create their first set of needs and preferences. If users already have devices configured, then a simple “Capture and Create” tool could be used to grab their settings and create their first Needs & Preferences set. If they cannot or do not currently use devices, then a

“First Create” tool might be used that assesses what is needed before the person obtains access, and then uses that information to create their first Needs & Preferences set. They can then go on to **Explore** tools to fine-tune their preferences. A “Preference Management” tool could be used to create a first Needs & Preferences set, but would require advanced skills and knowledge about a user’s needs and how the preference choices they elect apply to the system.

- **Exploration Tools** provide an **Explore** (and a simple kind of **Manage**) function; they are used to allow users to explore different possibilities have been developed with the goal of encouraging users to dive more deeply into the process of preference profile development. These tools introduce users to new or unfamiliar ways in which content can be adapted, while also encouraging those who are familiar to try new transformations. A successful exploration will assist users in exploring adaptations without fear of permanently losing the current level of access. This means the tool must allow users to back off of a preference adaptation. It must also offer the option to play with different adaptations and let users decide if a given change improves the level of access or comfort with content. Once a user has reached a point at which he or she is comfortable with the preference adjustments that have been made, the user can choose to save these preferences for use in later contexts. At this point a more advanced user may enter a “Preference Editor” tool. Alternatively, the user can choose to use the current preferences only within their current context. With each of these potential implementations, it is important to customize these tools to fit the expected context (e.g. age of users, subject of materials being used) and the expected outcome.
- **Preference Editors** provide **Adjust** or **Manage** functions and allow users to declare, modify, and apply preferences across different contexts and devices. These tools are useful for users who are already familiar with the concept of digital accessibility preferences generally, their own preferences specifically, and with the ways in which these preferences can be accessed and expressed in any given digital contexts. Preference Editors allow users to have access to all of their preferences in a centralized tool where users are able to continuously update and manage preferences. These tools can range from simple to complex so that the tools themselves can meet the Needs & Preferences of different users.

Importantly, each tool in the ecosystem shares a focus of empowering users to discover, explore and adjust user interfaces to meet their needs, but the tools vary in exactly **how** they accomplish this, **when** they would be used, and the required **skill levels** of the users. Furthermore, there is no one single implementation for each tool—there should be many implementations that engage different types of users and/or that are designed to be used in different contexts.

In addition to the above, the ecosystem also comprises a Personal Control Panel (PCP), which seeks to provide convenient, instant access to a user’s selected set of feature settings on the device (not preferences, but real-time settings). However, it is also convenient to put some controls on the panel that would allow Capture of settings or to launch Preference Tools. So while it is not a preference tool, it is tightly integrated into the Preference Tool Ecosystem.

In understanding the tools (especially the PCP), it is important to distinguish between settings and preferences. By **settings** we are referring to the current state of the settings on a device or software package. Examples include the current volume, speech-rate, and status of captions (on or off). By **preferences** we mean the **stored** preference that a user has for some setting. For example, they may prefer that the captions be on whenever available, or they may prefer that captions be turned on whenever the volume in the room is above a certain level. These are preferences—or desires that are stored somewhere. When the preferences are applied to a device or software, they change the settings of the device or software to what the person prefers. If we design this correctly however, users will not need to make this distinction. They can think of them all as settings that are saved and applied the way they prefer and when they prefer.

5.2.3.2 Defined Specific Roles and Design Constraints for the Preference Tools

The section that follows gives a brief description of each tool defined in our work, and outlines the user activity spaces, the interaction methods that can be used, and the special constraints or requirements of the tools to enable users to interact and accomplish tasks within these spaces.

First Discovery Tools

The role of the First Discovery is to find out what the basic needs of a person are, needs that, if they are not met, would prevent the person from using the ICT at all. First Discovery and addressing access to single sign form the key components necessary to getting a user “in the door” to any ICT system and then on to any preference gathering tool to enhance their access. Until we know these basic needs we cannot begin to give the user an **Explore** tool because we do not know what would be within their ability to perceive or understand. Most users who use a First Discovery tool will then be guided toward a tool to Explore. First Discovery tools are only used with people who do not currently use ICT and therefore do not already have settings or AT that can be captured and used as a starting point.

A First Discovery tool is probably one of the most challenging tools since it must start without any assumptions about the user and slowly establish what types of input and output a user is able to interact with, before moving into a space of preference setting activity.

Within First Discovery, all of the underlying complexity of the preference exploration and preference management tools exists. However, additional complexities must also be accommodated. These include: determining what can be used for input or response from the user when the user cannot operate one or all of the traditional computer input devices (keyboard or mouse or touchscreen), and determining what can be used to present information to the user when the user cannot perceive or understand one or all of the traditional computer output (audio, video, braille display, etc.).

First Discovery must therefore employ a different strategy from the other preference gathering tools. With First Discovery we need to turn on many of the accessibility features, critical to getting the user in

the door, such as text to speech and then begin to systematically remove settings. Other preference tools are more often than not additive, enhancing existing settings as the user goes through the preference gathering process.

We expect First Discovery to primarily rely on the following interaction methods:

- **Prompt & Ask:** First discovery tools are particularly challenging because they must have an interface that is operable by everyone no matter what their disability or combination of disabilities. This means that they must have parallel presentations (visual, auditory, tactile) and accept a wide variety of different response modes (voice, keyboard, touchscreen, pointing device, alternate input devices), especially in the initial moments. The system may need to try several different types of prompts or questions. For example, this might mean presenting an initial prompt in textual, audio, and visual formats (icon and sign).
- **Infer & Recommend:** Based on user input in response to prompts, questions, tasks and games, we expect the systems will need to infer which presentation modes are perceivable and understandable and usable for the user, but constantly check and confirm any inferences. Inference can be very important in helping to guide the discovery tool in efficiently trying different strategies in the early stages where effective communication with the user has not yet been established. The tool then needs to adapt all of its inputs and outputs to those modes

The key goal of this First Discovery process is to determine the limits beyond which the individual cannot perform (e.g., font sizes that will be too small for them to read, keys that are too small for them to hit, and controls that they cannot operate), and a rough estimate of their preferences along these basic parameters (e.g. approximate font size they like).

From here the user would likely proceed into an exploration tool to refine their preferences. They may do this directly or after they have had a chance to use their new access for a while.

Capture and Create Tools

As discussed previously, users who are already using access technologies (AT or built-in access features) can more easily create a first Needs & Preferences set by simply using a Capture and Create tool that would capture their current settings and preferences and create their first Needs & Preferences set from that. There is no need to discover what will work for them because they already have at least one set of preferences they can use. From there they can move on to the **Explore** tools to find out if there are better settings for them.

Exploration Tools

Exploration Tools provide a means for users to explore ways to personalize their experience in the context of a task they are performing. They are intended to expand the user's understanding of the

possibilities of personalized interfaces. Different exploration tools can support exploration and experimentation for different groups (by age, interest, ability, subject matter, task). We anticipate that the ecosystem will provide these contextualized tools, and that the role of the PGA project is to facilitate that by developing toolkits and providing exemplars.

Exploration Tools allow users to explore and play with content transformations and adaptations in context. Exploration tools may provide their own context, or they may allow the user to choose their own context (web page or task) and explore the effect of the different settings on that context. To operate in this latter form, a tool would need to be built so that it can be integrated into either web based or native applications.

As a part of our work in this area – we created (designed and developed) a working prototype Exploration Tool (<http://build.gpii.net/prefsEditors/demos/discoveryTool/>). This tool provides users with presets, which are logically grouped selections of preferences with default settings. This prototype tool is an example of what tool developers can easily do. This prototype (as well as the design exemplars in Appendix A) is a good starting place for tool developers to easily see, modify, and adapt for use within other contexts. For example, a museum curator could easily use these tools as a starting place to design and develop a tool for use on a museum kiosk.

The current design sketches for the prototype Exploration Tool support the user in the following activities:

- Explore: Users can turn presets on and off, and adjust settings. The user will see these adjustments immediately enacted on the content they are viewing. Additionally, users can refine presets by adding custom preferences or removing default preferences, or try out random sets of presets as a way to explore the possibilities available to them.
- Adjust: Users may also undo any individual action, as well as set all settings and preferences back to default.
- Store: Once the user is satisfied with the effect of a particular preset he or she can save this setting locally or save the setting to the cloud.

The design sketches for the Exploration Tools illustrate the following interaction methods:

- Prompt & Ask: The tool prompts the user to “Try Something New” and asks the user if he or she would like to “save” the new preference.
- Show: Users are shown presets for “contrast”, “enlarge”, “more words,” and “speak,” as well as adjusters that they can use to adjust size, contrast balance, and rate of speech.
- Infer & Recommend: The tool can suggest preferences or settings based on user actions such as frequent error correction of multiple keystrokes.

Preferences Management Tools (PMTs) and Mini-PMTs

Preferences Management Tools (PMTs) are web-based tools that give users the ability to see and edit their preferences for any device, application, or context. As a concept, these tools are web based to provide cross-platform compatibility, ease of adaptability, and re-use of code. These are robust tools that provide the user with the ability to customize preferences for one or many devices. They allow the user to change and apply global preferences or define and edit preferences for different specific devices and contexts.

The current exemplar PMT design sketches

[\[http://wiki.fluidproject.org/display/fluid/%28C4A%29+Preference+Editor+frame+mockups+for+iteration+2\]](http://wiki.fluidproject.org/display/fluid/%28C4A%29+Preference+Editor+frame+mockups+for+iteration+2) support the user in the following activities:

- Adjust: Users can adjust settings for text size, cursor size, contrast, and text to speech, and access previews of how these adjustments will appear on their devices.
- Store: Users can import settings from existing devices, and choose to apply entire sets or specific preferences to new devices. Users can also create new preferences and choose to apply these to any of their existing devices. These settings are then stored in the cloud for future use.
- Manage: Users can add or remove contexts and devices (i.e. home or school and phone or desktop), create preference sets for different contexts and devices, share and/or sync preference sets with other users, and provide feedback to the tool.

The PMT also illustrates the following interaction methods:

- Prompt & Ask
- Show
- Infer & Recommend

The current design sketches for the PMT represent one version of what a Preference Management Tool might encompass. Not all PMTs will necessarily provide support for all of the user activities listed, and may use any combination of the interaction methods listed. Some will be more complex and powerful for those who want very tight control over preferences. Other PMTs could be very simple and provide more limited functionality.

Personal Control Panels (PCPs)

Personal Control Panels (PCPs) provide users with an easy means to directly adjust the device's/software's settings on the fly. When users create their preference set, one option they will have will be to have a personal control panel for quick adjustments to the access settings on their device(s). This tool would always be present on a device. A PCP allows the user to quickly choose "for the moment" settings they may need from a short list of features they feel the need to have quick and constant access to. (This list of settings can be chosen by the user from her preference set, or pre-

populated with settings the user might find useful). PCPs can also give users the option of saving settings for later use on that device or on other devices. That is, they can have preference saving capability – or they can launch a preference tool to do this.

- PCPs are PANELS (visual or auditory) that contain CONTROLS where the Panels are PERSONAL (that is, each person determines what is in her Personal Panel).
- PCPs key role is to allow user to control the settings of the device they are currently using (e.g. turn captions on/off right now -- or change speed of screen reader right now).
- PCPs are (usually) applications local to a device (so that they allow instant control and are not impacted by network delays or connectivity).
- the PCPs for a single person can be different in different contexts (device, applications, etc.)
- PCPs can launch Preference Tools (e.g. capture, adjust, explore, or manage tools)
- A PCP could also have an embedded PMT (at this point the PCP blurs with a mini-PMT) though it would usually launch one since PCP usually need to implemented as native application or OS extension and PMTs are usually web based (for multiple reasons)

PCPs are launched (and configured) in the same way as any other AT or access features in the GPII. That is, it is created and configured by the GPII launchHandler and settingsHandler on the device based on user preference (default preferences for the PCP or preferences for PCP based the current context). This will be discussed more below in the architecture section.

5.2.3.3 Defined Ways that Inference Can Be Used Within the Preference Tools

We identified key ways that inference could play an important role within almost all of the tools outlined above:

1. **To guide users efficiently through the preference space.** For example, the mouse speed preference is highlighted after inferring a user’s difficulty aiming with the mouse, or keyboard preferences are suggested based on typing errors that have been observed. If it is determined that a person cannot see the screen at all, there is no need to present font face, line spacing, or other visual preferences.
2. **To suggest specific preference settings for users.** For example, users are offered the chance to try a scan delay of 1.2s, instead of asking what scan delay they prefer.
3. **To tailor the tool’s user interface** to maximize the user’s control and understanding while maintaining simplicity. This is especially important in First Discovery tools. For example, if we infer from lack of response that a user has physical trouble using some input methods, the user is able to explore this with simple exercises to determine which types of controls/input he or she can physically operate, and those controls/inputs are used for the remainder of the tool interaction. Similarly, a tool might infer, from difficulties the user has with certain tasks or exercises, that he or she has trouble with, e.g., that type of control, with complex instructions, or with memory, and then tune the session to explore presentation strategies that work.

Inference can also be used to suggest an alternate physical entry device be tried. There may be cases when suggestions may extend beyond preference options to recommendations of hardware that could assist the users. For example, an alternative input solution may be necessary if a user is operating a tablet but is unable to slide open a hidden window through the use of touch.

When users are working within a context where preference setting is not their primary activity, inference can provide information to be used later when they turn their attention to preference setting. (User permission must of course be secured for any monitoring activity.)

Inferred preferences can be drawn from several sources, including the following:

- Measurement or evaluation of test-specific user actions (e.g. activities within a PMT or First Discovery Tool)
- System evaluation of in-context user actions (e.g., repetition of a single character when typing in an educational site that includes a Discovery Tool)
- System evaluation of completion time for specific tasks
- Statistical analysis of performance based on applied settings
- Context or user environment (e.g. location, device being used, etc.)
- Current device settings
- Preference data from peers (e.g. inference of preferences from a statistical analysis of the user's profile, in comparison with other profiles)
- Transformation of clinical data into inferred preferences

Inferred preferences that have been made available to a tool can be used to guide the tool interaction. For example, if a user has made clinical data available, and a set of potentially useful preferences has been inferred from this data, a PMT or First Discovery tool interaction can start with these suggested preferences, and the tool's interaction mode can be tailored. If the clinical data suggests a specific minimum target size, the tool can use that size of targets (or larger) until the user's preference has been established, and the target size preference can be given priority within the tool. Research is needed to establish methods for a PMT or First Discovery tool to manage and prioritize the set of possible preferences, in order to guide the user more efficiently through the preference space.

Confirming with users is natural in the context of a PMT or First Discovery tool, but leads to a potential pitfall in a Discovery tool, where preference setting is not the user's primary goal. An implementation that interrupts users to make suggestions would be intrusive, and should offer a way to turn off further suggestions. As an alternative to interrupting the user, inference can be used in a Discovery tool as a way to identify appropriate presets to offer when the user chooses to explore the tool.

The decision on what preferences to prompt the user with can be informed by presenting simple activities and measuring performance. Once the user performs the activity, specific settings can be suggested and tried out. For example, if a user is requested to type the word "cat", and the typing is

analyzed, then the keystroke timing and errors can suggest changes to the keyboard response settings that would improve accuracy. If an individual presses the same key twice in quick succession, the timing of these key presses gives a strong indication of whether the second press was deliberate or accidental (Trewin, 2004). From this it is possible to infer whether the BounceKeys feature would help to eliminate unwanted extra characters. Most users are not aware of this standard keyboard accessibility feature, but would be able to decide whether it was useful by trying it out and seeing the effect on their typing. After inferring the need for BounceKeys, the tool would provide the chance to try it, and provide a description of the proposed change. The user could accept or reject the proposed new setting based on a trial. This makes the interaction accessible to people who have limited knowledge, literacy, or technical skill. We believe that the success of this scenario is predicated on the system displaying the feature to the user based on inference, rather than waiting for the user to first somehow identify it as a possible preference. We do not believe that this violates the user's autonomy.

Another example of inference could provide live mouse targets of different sizes and let the user experiment with the choices and indicate his/her preference. An inference-based version would observe, in the background, user performance with mouse targets such as accuracy, overshoot, time on task, and prepare a candidate preference setting for submission to the user.

Tool actions based on inferred preferences must be implemented with awareness that the inference may be wrong. Thus, it is not appropriate to change active system settings without both making the user aware of the change, and providing a way to reject the change. This also applies to the tool's own interaction method. For example, if a First Discovery tool infers that switch access might be better for a user, it cannot simply switch to a scanning interface – both scanning and direct access options could be offered. On the other hand, if the tool infers that the current targets are too small, the user may be unable to continue until larger targets are provided. Managing this interaction in an accessible way is a significant challenge, and is an important area for further work.

5.2.4 The Preferences Framework: Architecture Design

5.2.4.1 Identified Key Architecture Design Constraints

A key development of the first year has been the creation of the Preferences Framework, the Preferences for Global Access architecture. This architecture builds on the Global Public Inclusive Infrastructure (GPII) by providing tools to support the creation and integration of tools in the Preference Tool Ecosystem into a variety of web-based applications, content management systems, and delivery environments. The PGA work has focused on providing extensions to the GPII architecture that support the creation and management of preferences. The Preferences Framework is designed as a rich set of technical building blocks and modular and reusable tools for the development community that can be used to create a wide variety of different (but compatible) tools to meet the needs of the different users and contexts in the Preference Tool Ecosystem. This enables the Preferences Framework to support

customized user interfaces, adapted web content, and native accessibility solutions that meet an individual user's needs in a reliable and predictable way across all their devices and operating contexts.

This Preferences Framework is designed to also support a variety of privacy and security techniques required to protect personal information. Privacy and security are foundational concerns for this architecture, which is built on the principles of minimizing inadvertent information leakage and empowering users to define their own networks of trust when sharing preference information with third parties.

Reusable Components

The Preferences Framework is designed to support developers by providing reusable customizable components that enable the creation of one-size-fits-one solutions that best serve the diversity of users in the ecosystem. The Preferences Framework aims to make it easy for designers and developers to create different user interfaces for different users.

A primary concern of the Preferences Framework was to ensure that there was technical consistency and reusability across all of the tools development in the Preference Tool Ecosystem. In this resource limited field, it is important to avoid the costs of redundant or duplicative efforts—as well as the potential confusion for users—that typically result from ad-hoc software development. Developers of preferences editing tools need to be able to pool their efforts and share a collection of:

1. Flexible user interface components for exploring, declaring, editing, and managing preferences
2. Pluggable personalization services that do the work of transformation and delivery in order to allow users to instantly see/hear the impact of changes to their preferences
3. Tools that can configure web-based and native platform accessibility features and third-party assistive technologies, building off of prior PGA team member work on the Cloud4All and Flexible Learning Object Environments (FLOE) work
4. Robust and scalable programming components, frameworks, code modules and techniques that facilitate extensibility, user experience adaptation, future innovation, and sustainability of the system

To this end, the Preferences Framework offers a reusable set of schemas, programming APIs, and UI building blocks. These components are being employed in the development of several preference tools in the broader GPII ecosystem, including the PGA Exploration Tool prototype and the Cloud4all Preferences Management Tool (PMT) and Personal Control Panel (PCP). The implementation of these components in development will be discussed in detail in Section 5.2.3.2, Framework Components.

Adaptive User Interfaces

A key goal of the Preferences Framework approach is to help developers create applications that are designed from the beginning to be adaptable, customizable, and extensible rather than trying to retrofit or transcode inaccessible applications and content after the fact.

The PGA project team has extensive experience in delivering adaptive user interfaces designed to enable customization to meet the needs of individual tool developers, and the user communities for which tools are being developed.

Building adaptive applications requires a different approach to software development. Instead of hardcoding a single user interface, adaptive applications need to support flexible connections between modular user interfaces and underlying services.

When applications are composed from modular pieces and the connections between these modules are flexible, it is significantly easier to:

- deliver a transformed or augmented version of the user interface,
- connect up an entirely different user interface,
- reuse portions of an application in different contexts, including those that are situational (e.g. detection of context changes such as ambient noise and lighting, etc.), and
- extend or alter the features of an application.

Several modular components have been developed or refined as part of the PGA prototype to demonstrate how an adaptive approach would look in a preference gathering process including support for selecting a preference for:

- text to speech
- high contrast
- larger text, line spacing, and inputs
- synchronized captions and transcripts
- table of contents
- layout simplification

However our work on preferences has also identified the need for many other components that have not been developed in the broader GPII ecosystem. While these features don't represent the core focus of work, we note them here for reference given that many support the gaps in cognitive access identified as an opportunity space through our research. They include:

- highlight and speak
- dictionary and thesaurus
- word simplification
- grammar check
- spell check
- text simplification

- alarms, reminders, and notifications
- math rendering, navigation, and verbalization
- breadcrumbs and other navigational support
- group formation and communication on a task

The use of modular components, along with the Preferences Framework API with which others can create new components, will serve two key interrelated goals. First, it will reduce the cost, complexity, and specialized knowledge required to build adaptive applications; and second, it will help foster an open ecosystem of alternative user interfaces and delivery strategies that are tailored to different contexts and applications.

Web Based Content and Native Accessibility

The Preferences Framework employs a “web-first” approach, using standard web technologies wherever possible. This helps to ensure the broadest possible compatibility with different platforms and devices. It also lowers the cost of developing user interfaces and infrastructure by reducing the amount of platform-specific code that must be written. The [motivations for this approach](#) are described in further detail in the GPII wiki.

Over the past several years, the capabilities of the web platform have expanded substantially. Features that were previously only possible in native applications are now available to web applications, such as advanced graphics, audio, camera support, touch and device orientation input, text to speech, and more. Indeed, new operating systems such as Firefox OS, Windows 8, and Chrome OS support full-featured application development entirely with web technologies. As a result, the web platform is a potentially significant source of new assistive technology development in the coming years. Traditional “monolithic” AT such as screen readers are already available in web-based tools (e.g., [WebAnywhere](#) and [ChromeVox](#)). More importantly, the web is enabling the creation of modular assistive technology “bits” (such as the Enactors that accompany the Preferences Framework) that can be assembled into novel applications in the Preference Tool Ecosystem.

Nonetheless, not all user preferences can be supported in a purely web-based environment, including some of those in the [Preferences Matrix](#) documented in PGA deliverable 1. This is due to the fact that web browsers implement a strict “sandbox” security policy. A web application cannot modify the settings stored on a user’s computer. This policy helps to protect users from having their system changed by malicious or poorly coded websites. As a result, some features need to be configured and activated at the operating system level. For example:

- enabling system-wide high contrast mode
- launching a dedicated system-wide screen reader
- manipulating the keyboard response rate (for non-web content)
- interfacing directly with hardware such as switches or eye trackers

Due to these constraints, the Preferences Framework relies on the [GPII Flow Manager](#), which is the GPII component responsible for configuring the settings and assistive technology on a local device. The Flow Manager includes a collection of pluggable, operating system-specific [settings handlers](#) and [lifecycle actions](#) that are able to configure and activate settings such as the ones listed above. To do so, however, the Flow Manager must be installed directly on the user's desktop computer or mobile device so that it can interact with the operating system and native applications.

This lack of a built-in way for a web application to communicate directly with the Flow Manager presents a challenge for web-based preferences editing and discovery tools. These web-based tools need to provide users with the ability to try out and preview the effects of a setting before saving them. In order to accomplish this for all preferences, a communications channel is required between the web browser (where the preference exploration and management tools are) and the locally installed Flow Manager.

In the past, browser plugins have been used to bridge the gap between the web platform and the native operating system. In recent years, however, plugin usage has decreased, perhaps due to a combination of growth in the capabilities of the web platform along with a tendency for poorly written plugins to cause incompatibilities and security issues. Another disadvantage of the plugin-based approach is that each browser vendor has a different plugin architecture and API; consequently, separate plugins must be developed for each browser. As a result, our goal is to avoid browser plugin technology wherever possible.

A recent feature of modern web browsers is HTML5 [Web Sockets](#). Web Sockets are designed to allow a web application to communicate with a server using sockets. Notably, Web Sockets allow for "messages to be passed back and forth while keeping the connection open." In this way a two-way (bi-directional) ongoing conversation can take place between a browser and the server" ([Wikipedia](#)). This API is now included in today's browsers.

In keeping with the GPII's web-oriented approach, the Flow Manager has already been implemented as a web server, even when it is installed locally on the user's system. For security reasons, the Flow Manager server is typically kept behind a firewall so that it only accepts local requests. This nonetheless allows a preference tool running in the user's browser to use Web Sockets to communicate with the Flow Manager. Tools can thus instruct the Flow Manager to initiate all or part of the [personalization workflow](#) so that the user can actually try out new settings in practice. This solution will likely work in [most modern web browsers](#).

Although the core infrastructure is already available in the Flow Manager today, in order for the Web-based Preference Tools to operate properly, we have identified that further development work will be required to:

- provide a reusable Web Sockets client library in the Preferences Framework
- extend the Flow Manager's available RESTful APIs, exposing *preference set change events* that can be called by a web application via Web Sockets

Privacy and Security

Privacy and security must be principal tenets of a preferences acquisition system. Users want to have control over who can access their personal needs and preferences; our systems must protect this potentially sensitive information and give users easy-to-understand ways to identify their trust networks. When personal information is provided, an effort must be made to ensure those preferences do not reveal their medical information.

Security support must also extend beyond the protection of a user's information to address security walls laid down between the web browser and the local system. As such, the Preferences Framework outlines security considerations both for users preferences and to address security issues between the browser and native platforms.

Security and Privacy of User Preferences

A user's preferences may reveal personal information about their lives and needs, either directly or indirectly; privacy and security is thus of critical importance to the design of the Preferences Framework. This is another case where the project will closely follow the approach established by the GPII. While this security and privacy infrastructure work is still in its early stages, the intention is to use the OAuth 2.0 framework to protect access to a user's preferences by third party applications.

The use of OAuth will let users approve or deny access to their information on a per-site basis, ensuring they control who gets to see and act on their needs and preferences. In designing this system, the Preferences Framework will strive for compatibility with efforts such as the National Strategy for Trusted Identities in Cyberspace (NSTIC), a White House initiative focused on the establishment of an identity ecosystem that will allow individuals to validate their identities securely to facilitate exchange of personally identifiable information such as accessibility preferences. This layer of protection is particularly critical for the exchange of data indicating medical needs of users (such as the need to use a screen reader) in contexts where that information may be exposed to third party applications unbeknownst to the user.

In the future, an attribute release layer such as the [Kantara Initiative's User Managed Access](#) will be layered on top of the basic OAuth 2.0 authorization system (Machulak 2010). This will give users the further ability to specify that a site or application is able to see only portions of their preferences set, ensuring that the risk of "privacy leakage" is reduced by only sharing the minimum information required by a service to meet users' needs (Krishnamurthy 2009).

The goal is to ensure that the architecture can support diverse privacy and security requirements internationally.

Security Between the Browser and the Native Platform

As mentioned previously, web browsers protect the user from malware by “sandboxing”—not allowing direct access to features such as system settings or local files. Over time, standardized browser services are being added to provide access to some of these features securely. In the interim, developers sometimes make use of browser plug-ins to access features in the native platform or they embed the browser in containers like [Cordova](#) to gain access to these extended features on mobile devices as plug-ins themselves are not supported on most mobile browsers. Near term work on preference acquisition must work within these restrictions as direct access to the local features will be necessary. However, new trusted API services are being added to web browsers such as [WebSockets](#) and [IndieUI User Context](#) that allow access to local information either directly or indirectly. Future preference acquisition and manipulation solutions will make efforts to use these services to reduce management costs as we increase portability of our solution.

Assistive Technologies and Security

At times activating native assistive technologies may be required in order to acquire a user’s preferences. Many assistive technologies can be mistaken by the operating system as malware, in that they must access native operating system services that would not normally be accessed by native platform applications. For example, screen readers and screen magnifiers still hook into graphics engine drawing calls and “steal” the keyboard away from applications in order to facilitate control of their assistive technology by the user. Consequently, on Windows these applications must follow [security requirements for Windows](#) such as being signed. On other platforms, such as iOS, assistive technologies with this level of system access are not allowed and we must be dependent on what the native operating system platform provides.

Additionally, we must also consider that during preference acquisition a user may be using a shared system. Because of this consideration, it is important to ensure that any preference acquisition tool is able to reset a user interface back to the original system state that existed prior to a user’s customization of the interface. In the GPII infrastructure, this process of “snapshotting” a system’s state prior to personalizing it and then restoring it afterwards is accomplished by the Lifecycle Manager component. The Lifecycle Manager is the part of the Flow Manager that is responsible for coordinating all aspects of the workflow of configuring, starting, and stopping assistive technologies and solutions on a local device. All settings that are set by the Lifecycle Manager are automatically reset to their previous values when the user’s session is finished. [Further information about the Lifecycle Manager](#) and its role in the GPII architecture can be found in the GPII wiki.

We must ensure that when native platform assistive technologies are involved in the preferences gathering that platform conventions and policies will be adhered to so as to ensure the security of the user and the system on which they operate.

5.2.4.2 Identified Preference Framework Components

The Preferences Framework is intended to provide the critical infrastructure needed by developers of preferences editing and discovery tools. The development of the Preferences Framework is informed by the practical, day-to-day development issues that arose during the project's prototyping efforts, as well as those of Cloud4all's in progress Preferences Management Tool. For example, developers of these tools found that they were writing overlapping and redundant code for managing preference sets, dynamically presenting user interfaces tailored to each user's preferences, and handling common tasks like internationalization and skinning. As a result, a collection of shared application programming interfaces (APIs) and reusable components were created by the PGA project that takes care of these tasks for developers.

What the Preferences Framework Offers

Today, the Preferences Framework offers components and APIs that address several key development activities common to all preference editors, including:

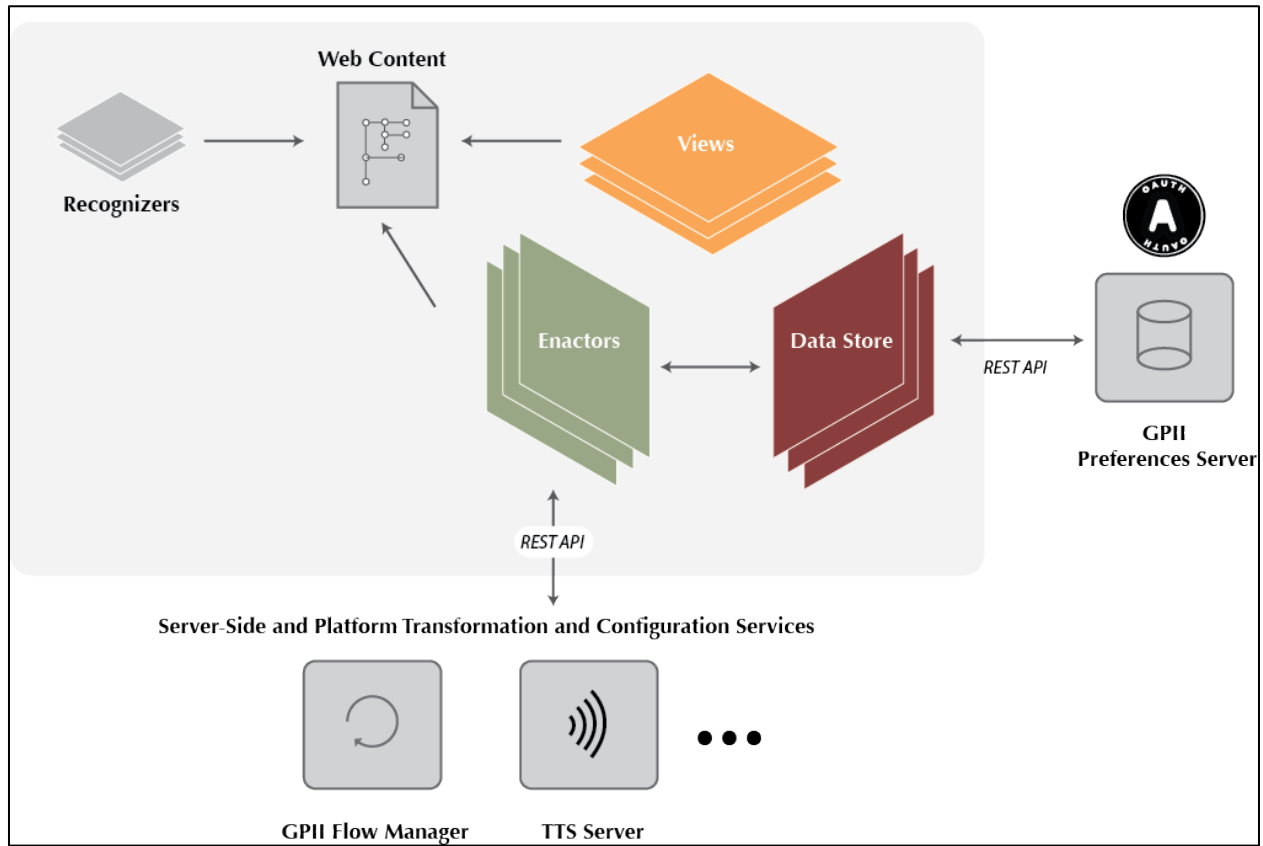
- saving, retrieving, removing, and versioning a user's preferences with the GPII Preferences Server and to the local device
- dynamically generating user interface controls that allow users to edit and refine individual and clustered preferences
- providing links to GPII "auto-personalization from preferences" (APfP) components to enact the user's needs and preferences by transforming web content and user interfaces, injecting enhancements and scaffolds into an application, and configuring and launching web, cloud, and native access features and services

Upcoming versions of the framework will add support for several other key features such as:

- managing multiple preference sets and contexts
- support for undoing previously-saved changes to preference sets
- sharing preferences with individuals as well as group collaboration
- inferring preferences and settings based on activities and metrics
- mechanisms for inferring information to guide a tool's interactions steps and methods used to communicate with users
- deeper integration with native platform services and applications

Figure 2, below, illustrates the components that are offered within the Preferences Framework, as well as some of the services and tools they interact within in the broader GPII ecosystem. The Preferences Framework offers: 1) Views and Adjusters, which provide reusable widgets and controls with which users can edit their preferences; 2) Enactors, which do the work of transforming the user interfaces based on these preferences; 3) Data Stores, which provide persistence for the user's preferences; and 4) Recognizers, which provide activities to help infer new preferences and settings. These components typically interact with both web content and a suite of native/server-side GPII transformation services, including the Flow Manager, which is responsible for configuring and launching native access solutions.

Figure 2: The Components of the Preferences Framework



Persistence

All preference-editing tools need to be able to store, retrieve, and keep track of revisions to a user's preference sets. The Data Store abstraction has been designed to provide a single programming interface for different persistence strategies, including saving to the GPII Preferences Server and to local data storage on the device.

Views and Adjusters

Users often need to refine and adjust their preferences. Views provide a generalized abstraction for rendering editable interfaces that can be bound to and illustrate particular needs and preferences. The Preferences Framework provides support for both individual and clustered preferences, ensuring that designers can tailor the user interface to the context and user.

Adjusters provide fine-grained control over individual preferences. An effort is currently underway in the Cloud4all project to create a common pool of Adjusters that can be used in both PCPs and PMTs.

Enactors

Enactors, as discussed above, do the work of transforming the user interface and content to suit a user's needs and preferences. These are essential for providing the ability to see the effect of settings on real web and native applications. They typically interact with web, cloud, or native services such as the GPII Flow Manager or a Text to Speech Server. Enactors typically require a deep integration with a web application or delivery system, and as a result there will be many different types of these components. Indeed, it is possible to consider the entire GPII native Flow Manager and its Settings Handlers as a collection of Enactors tailored to adapting desktop and mobile devices.

Recognizers and the Proposed Inference Architecture

As mentioned above, inference is often best done in the context of activities and games that the user participates in. This approach to needs and preference gathering has the potential to streamline and enhance the user's experience, though it offers its own set of complexities and limitations.

Given this, there is significant value in providing an extensible and reusable architecture for supporting inference. Such an architecture has been recently designed by members of the PGA project to fill a gap in the GPII architecture, although this design has not yet been implemented in the Preferences Framework. Further design and development of this inference architecture is a significant opportunity for future research and innovation.

Recognizers

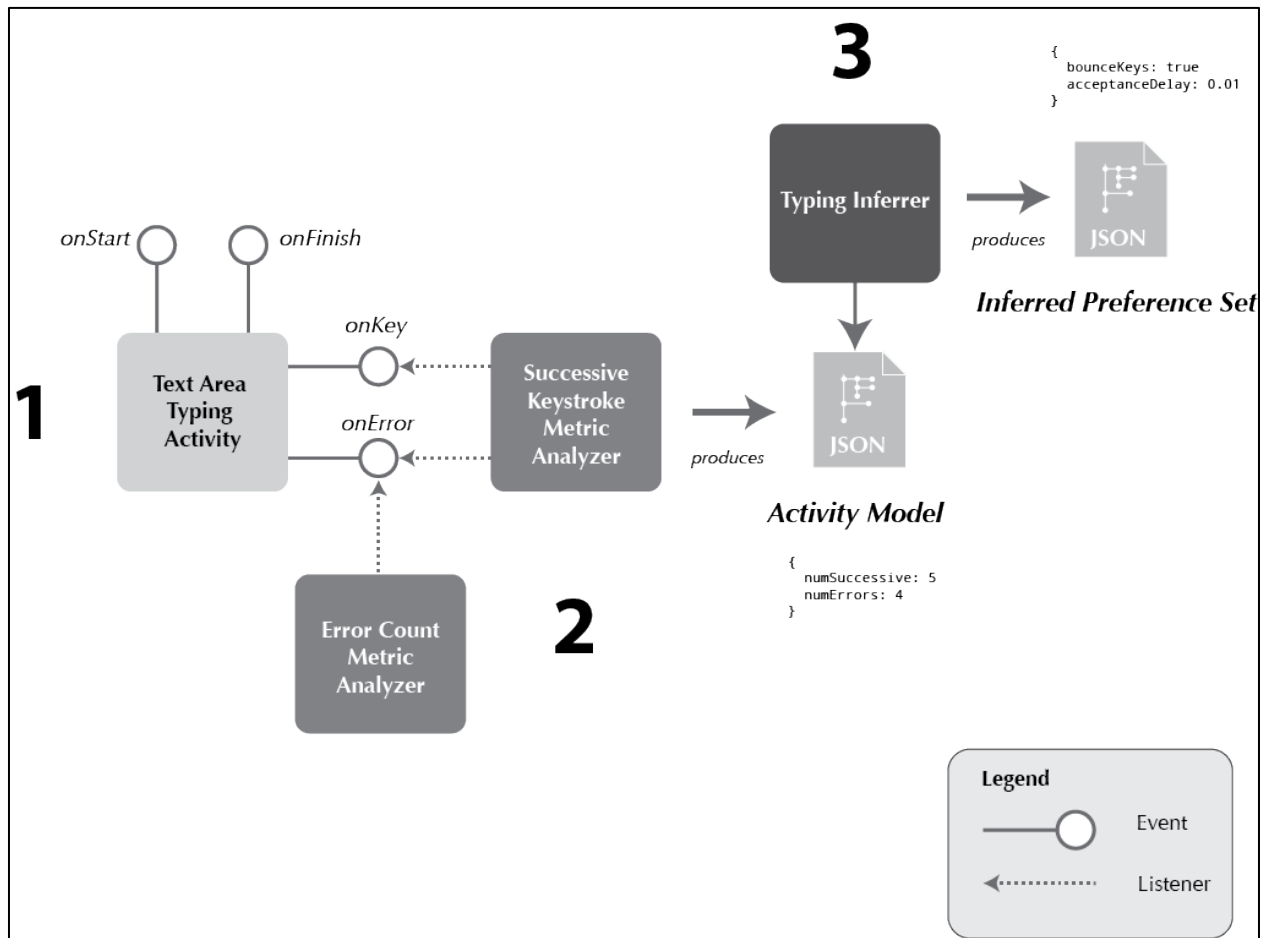
Recognizers are an architectural component that encapsulate several different aspects of the inference process, including:

- the sources of input and activities that a user engages in
- different strategies for gathering metrics analyzing user activity
- inference strategies, which translate metrics into a suggested preferences set

It is envisioned that developers will be able to create and share new types of Recognizers by composing together different *Input Sources* (also referred to as *Activities*), *Metrics Analyzers*, and *Inferencers*.

Figure 3, below, illustrates the event and model-driven relationships between parts of a Recognizer. Activities are a type of View component that provide a source of user input events. Metric Analyzers use these events to produce a JSON-based model of the behavior of the user during the activity, and Inferencers use this raw data to produce suggested preferences for the user. This architecture enables different approaches to metrics gathering and inference to be plugged into reusable activity UIs.

Figure 3. The Anatomy of a Recognizer



In order to clarify how the components depicted above interact, we will consider one of the inference examples described previously in this report:

“Given a demonstration of an individual’s typing, their keystroke timing and errors can suggest changes to the keyboard response settings that would improve accuracy. If an individual presses the same key twice in quick succession, the timing of these key presses gives a strong indication of whether the second press was deliberate or accidental (Trewin, 2004). From this it is possible to infer whether the BounceKeys feature would help to eliminate unwanted extra characters.”

An Activity is some kind of interaction presented to the user as a source for inference. In practice, this is often an exercise or game. In this example, it consists of a typing exercise designed to measure the user’s keyboard accuracy. Activities are typically composed of a user interface (a *View*, in architectural terminology) and one or more Input Sources, which connect with an input device of some kind (e.g. keyboard, mouse, etc.). The result of the user’s interaction is exposed to the system in real time as a

stream of events. These events usually represent higher-level concepts such as typing errors, goals or achievements, etc. They are connected to low-level input events from the browser or native hardware.

Metrics Analyzers

Metrics Analyzers do the work of listening for one or more user event and producing a data model consisting of metrics about the user's interaction with the Activity. These metrics are stored in a JSON object (a *Model*, in architectural terminology) that can be later processed and transformed by the rest of the inference pipeline.

Typically, a Recognizer will employ multiple metrics-gathering strategies. In our example, the “successive keystroke” analyzer will listen for keyboard events and keep track of the time between key presses. If keyboard events occur within quick succession and close proximity, the analyzer identifies this as a likely error and updates the model accordingly. Metrics Analyzers are also responsible for tracking and drawing metrics from temporal characteristics of the user's activity. As they listen for events they may choose to record data about how long it takes for the user to accomplish a particular activity or, as mentioned in the example above, the timing between particular events.

Inferrers

Inferrers are responsible for translating an activity model into a list of suggested preferences. In our example, the typing inferrer will suggest values for “bounce keys” and “acceptance delay” based on metrics such as the number of successive keystrokes and their timing. These preferences represent an “inferred set,” which can then be shown the user, who can confirm whether or not they want to include these inferred preferences in one of their saved sets. Showing the user may involve the use of different views and/or enactors to show the impact of the change and allow the user to try it in an effort to facilitate the user's decision-making process. Inferrers typically operate on multiple metrics.

5.2.4.3 Conceptualized Developer Tools

Rather than having to write a preferences editor from scratch, the Preferences Framework being developed provides a means to generate the user interface and persistence layers from a set of declarative (i.e. JSON-based) schemas that are assembled by a *Builder* into an editor that can be instantiated and presented to the user. The advantage of this declarative approach is that it is extremely easy for developers to borrow and adapt portions of an existing preference editor and extend them according to their own requirements, increasing the opportunity for sharing and code reuse.

Preference editors are constructed by combining three main parts:

1. *Panels*, which contain *Views*, *activities*, and *Adjusters* for users to edit their preferences with, and try out new preferences and activities.
2. A *Data Store* that provides persistence for the user's selections

3. *Enactors* that respond to a user's settings

The Preferences Framework provides all the lifecycle events, configuration hooks, and persistence infrastructure required to support the aggregation of these pieces, as well as providing some pre-existing panels and enactors that can be reused if desired.

5.2.4.3.1 *The Builder*

The starting point for working with the Preferences Framework is the *Builder*, a tool that creates all the necessary components (the settings store, panels and enactors) based on a set of schema files. Schemas define:

- the settings available to be edited
- the panels used to configure the settings
- the enactors that respond to the settings

The Primary Schema

The Primary Schema defines each of the settings that a preference editor will allow the user to adjust. This includes:

- their type (e.g. Boolean, string, number)
- their default values
- ranges, enumerations, and any other information necessary to define them, depending on their type

In future versions of the Preferences Framework, primary schemas will typically be generated by exporting information from the ISO 24751 AccessForAll Common Terms Registry, which provides an open forum for defining standardized preferences and settings.

Panels and the Auxiliary Schema

Panels are components that present adjusters—the controls that allow users to adjust the preference settings. Each panel defines a user interface (HTML, CSS, and JavaScript) as well as a preference map, which binds the state of the component (i.e. the panel's controls) to the primary schema. For inference related panels, the binding may be to settings or to some issue that needs to be addressed.

Along with the panel definitions, an *auxiliary schema* binds together the panels and enactors to be used for each of the preferences defined in the primary schema, such as an alternative panel with larger buttons designed to facilitate mouse interaction. It is how all of the component pieces of a preference editor get woven together to deliver a seamless user experience.

The Preferences Framework also includes a number of CSS stylesheets that are used to control the layout and appearance of the interface. These stylesheets can be used as the basis for flexible styling, and can also be overridden and extended as necessary.

Source Code and Documentation

The Preferences Framework is being developed as open source software that anyone can use and contribute to. So far, it has been used to develop the [PGA Exploration Tool prototype](#) as well as the Cloud4all Preferences Management Tool and Personal Control Panel. The code for all these editors is available [in a shared GPII repository on Github](#). This development effort is an excellent illustration of how PGA's research insights and prototyping activities have helped to foster collaboration and code sharing across communities and projects.

A tutorial that teaches developers [how to create a preferences editor](#) is currently being written and [reference documentation](#) is available for the preferences framework.

Cross-Platform Support

With the aforementioned web-first architectural approach, we are well positioned to support a very broad range of platforms and devices. Preferences editors and discovery tools, for example, will run on any platform with a modern browser. Supporting native access features and assistive technologies, however, requires deeper integration with low-level platform services that can be complex and highly variable across platforms.

The Preferences Framework is currently being developed for the Windows, Linux, and Android operating systems. To varying degrees, each of these platforms offer APIs and services that can be used to configure, launch, stop, and restore settings for both built-in operating system access features as well as third-party native assistive technologies. The architecture is fundamentally designed to run across different platforms, and we anticipate that future support for operating systems such as Mac OS X, Windows Phone, or iOS can be added. The Preferences Framework is coded entirely in JavaScript, which can be run on nearly any native platform.

Increasingly, many users are employing mobile devices and tablets. iOS has become particularly popular due to its excellent suite of built-in access features. The PGA and broader GPII communities are keen to support this platform, but it presents a number of fundamental and troubling limitations. Apple places intense legal restrictions and a rigorous app review process on developers. On iOS, apps are completely sandboxed—they are not allowed to access system settings or the settings of other applications. Any app that attempts to do so will be refused by Apple. Mac OS X, with its new GateKeeper feature, is increasingly taking this troublesome direction as well.

As a result, given limited resources and a desire to make an impact on users today, iOS is not currently supported for native application personalization. All web-based tools and access features are supported, however. Though it is technically possible to add support for Apple-based platforms to the Preferences Framework, users will be unable to benefit from it until Apple makes approved APIs for personalization available in their platforms.

6. Next Steps in the PGA Roadmap

6.1 Gather End User Input on the PGA Ecosystem and Framework

The critical next stage of PGA is to devise ways to gather input for the broad participatory engagement of end users. Broader end-user input is required to iteratively design, develop, refine and to evaluate the tools and editors in the Preference Tool Ecosystem and to extend the Preferences Framework. End users include:

- the large range of individuals who face barriers to accessibility,
- caregivers, family members and friends who provide support and guidance,
- assistive technology and rehabilitation professionals who bring professional and tacit knowledge regarding what works to address specific barriers,
- researchers whose focus of research is accessibility and who can contribute to the knowledge store of what works,
- individuals who are in a position to also assist end users in choosing and editing personal preferences, including librarians, teachers, community center staff, etc.,
- entities and developers who may develop new preference editors or extend existing preference editors using PGA software development toolkits.

The testing will ensure that the process of providing input to the PGA development process is accessible, understandable and efficient. The testing outreach strategy will include a recruitment process that reaches out to an ever-widening group of potential users including users that the team has not yet considered.

6.2 Develop Preference Tool Ecosystem and Preferences Framework Exemplars

In Year 2, the PGA team will further refine the Preference Tool Ecosystem and Preferences Framework to build exemplar tools and advances to the architecture specification that focus on the most difficult to design components in our landscape. To inform the development of developer ready software development toolkits, the team will create an exemplar First Discovery Tool for the Preference Tool Ecosystem. To further refine our Preferences Framework by tackling some of the trickier architectural

challenges around inference surfaced in Year 1, the team will further articulate a roadmap for inference architecture.

On the Preference Tool Ecosystem side in Year 2, the PGA team will develop, test and refine a First Discovery tool. This requires the very difficult task of designing an interface that speaks the language of the user without knowing anything about them. The tool will offer strategically designed basic choices and then responsively leverage the choices made as a launchpad or starting point for more sophisticated preference exploration, creation, refinement and management.

To further build the Preferences Framework on the architecture side in Year 2, the PGA team will build an inference architecture that showcases best practices of judicious and effective presentation of guidance derived from inference. The PGA team will work in close collaboration and coordination with the larger GPII community and projects to outline these guidelines and build a prototype that showcases them. Careful attention will be paid to the interoperability and integration of the inference support model with the larger GPII ecosystem.

6.3 Enable a Community of Designers and Developers

While the PGA team will develop a set of preference tools, the sustainability, inclusiveness and scaling of the preference system is largely dependent on contributions and participation of the broader community of developers and implementers. The PGA team will develop a software development toolkit that supports developers in getting started with the Preference Framework and Preference Tool Ecosystem. The toolkit will strive to ensure that the creation or extension of preference tools as simple as is optimal - allowing designers and developers from the growing community to focus on the new preferences to be expressed and not the technical complexities of code. This will be achieved, in part, by providing components that can be easily assembled and remixed. The growing pool of preference editing tools will provide diverse variants that can be adapted and extended. This toolkit will be accompanied by responsive and extensible tutorials and help systems.

PGA will also provide a model for supporting community mechanisms that match end user demand and interest in features with potential designers and developers with the knowledge and skills needed to create new preference tools. This will help to create a sustainable system that is not constrained by levels of direct funding. To ensure that the available pool of developers with the knowledge and skill needed to create new preference tools widens, Year 2 will focus intensively on providing critical documentation and training supports needed to expand the pool of developers in the network.

6.4 Develop an Extensible Knowledge Store to Guide Outreach

All implementations of the AccessForAll standard have leveraged the collective knowledge and resource production made possible by networks. Not only are networks used to access repositories of resources, tools and derivatives but this principle also holds true for preference discovery, declaration and

refinement as well. If strategically designed, these networks can be used to suggest, test and refine approaches and resources for an ever-growing set of preferences. If a variety of participants, experts or knowledge sources are supported in reporting and describing things that work and the conditions under which a particular approach or configuration works, then these can be made available for selection, experimentation and refinement by other end users.

Given that the range of possible preferences can never be fully predicted, a system that is self-correcting, with the opportunity to report new approaches or preferences, and structures for continuous feedback loops regarding the success or failure of a particular match, will ultimately be much more inclusive than a static or episodically updated set of selectable preferences.

Given these insights, Year Two will place key emphasis on ensuring that dissemination of the outcomes of Year One are apparent in a call to action format that fully engages networks as a community of practice. This will ensure that new additions to the inventory of existing preference gathering tools and innovations are inventoried and tracked in an ongoing way, there is support in the network for tackling challenges in the development process as they emerge, and new voices join the mix of stakeholders engaged in this work.

7. Conclusion

A key finding from Year One of the PGA work is that any set of guidelines, and any list of prescriptions that assert the suitability of a preference setting for a given access issue is likely to change, will vary from individual to individual, and will vary for the same individual depending on the context or the goal. This is especially the case in less explored areas such as cognitive access, or as yet unexplored areas such as mental health and emotional access. Any preference framework or ecosystem must be extensible and avoid fixed, prescriptive decisions or assertions; it should facilitate the complex diversity of user requirements and the discovery of new approaches or strategies to address barriers and optimize accessibility. An effective framework must support iterative exploration, discovery and refinement.

At the same time the system must acknowledge that the amount of time and cognitive energy users can devote to the process of determining and declaring their personal preferences sets may be very limited or episodic, and that the range of insight and comfort with digital systems is a broad spectrum. The system must therefore be able to tailor the process to the situation of each individual and must make the process as efficient and effective as possible. Each individual should be able to choose the timing, granularity and degree of control of the preference selection process.

Building on these insights, the central output of our work has been the creation of the Preference Tool Ecosystem, which is a roadmap for flexible software development kits to support users in discovering and saving preferences. As part of this, we developed an adaptive model that places users at the center and employs novel methods for engaging users in a preference discovery process. These approaches include methods that aid users in learning how they learn best, a strong inventory of best practices for

inference driven approaches to preference acquisition aiming to make the learning to learn process more intuitive, and conceptual frameworks that outline ways of thinking about interface methods and user activity spaces that can be utilized to understand the ecosystem and guide tool development.

Based on our growing understanding of the Preference Tool Ecosystem, we also developed the Preferences Framework, which outlines the architectural specifications needed to support interoperability between the kinds of tools that would inhabit the ecosystem. The Preferences Framework ties new design thinking for preference acquisition to existing GPII architecture, and defines extensions to the GPII architecture to fill gaps needed to support preference inference.

In sum, the PGA team has created a Preference Tool Ecosystem and Preferences Framework that takes a functional approach to determining personal preference sets, acknowledges the importance of self-awareness and self-determination while also facilitating guidance from a variety of knowledge sources. These models acknowledge that there can be no one authority or initiative that can support this hugely complex domain and that any system or infrastructure must support collaborative, collective, distributed input and development in order to be sustainable and scalable.

The next stage of the project requires the engagement of the broad spectrum of users, potential developers, knowledge sources and implementers in the participatory design and development process. In collaboration with other GPII efforts, the PGA initiative will create the necessary architecture to support the creation of the ecosystem of preference tools delineated in the first phase of the project and PGA will also create exemplars and models of the range of tools that can be implemented and tested with a special emphasis on first creation and inference.

The first stage of PGA has made a substantial contribution to realizing the full vision of AccessForAll and the Global Public Inclusive Infrastructure. The work of PGA represents an essential building block of the overall GPII and AccessForAll ecosystem. PGA has iteratively researched and developed a design and prototype for a scalable Preference Tool Ecosystem that will serve the full diversity of users, their broad spectrum of goals, in the many contexts in which they require accessibility. The next step is to iteratively implement and evaluate this design with the full participation of people with disabilities and the broad support community.

Appendix A: Use Cases

As part of the project's user-centered approach the team wrote a number of detailed use cases and accompanying scenarios for specific tool interactions to explore and illustrate the concepts under development. The use cases and accompanying scenarios, [available as an external Open Author Resource](#), highlight several broad areas of consideration of user needs that emerged in our work including:

- Users' understanding of technology concepts and metaphors (including cognitive barriers, prior experience with and attitudes towards technology, etc.)
- Users' motivation and comfort in exploration and self-assessment
- The importance of technology to the user (i.e., frequency of use)
- Whether a user is acting alone or in a social context

Within each of these categories there is an extensive range of possible user needs and preferences. The use cases helped to highlight users' backgrounds, contexts, and goals, and allowed the design team to explore how these factors inform users' interactions with preference management tools.

Appendix B: Dissemination of the PGA Work

The PGA team has initiated a number of efforts to disseminate the PGA work. Each of these efforts—ranging from conference presentations, to workshops, to conversations with field experts, to web and social media dissemination—is detailed below.

Conferences

- **South by Southwest Education Conference** (March 2013) - Megan Simmons (ISKME) and Colin Clark (OCAD) presented the vision and objectives of the PGA work in a presentation titled, *Portable Profiles for High Access Learning*.
- **International Technology and Persons with Disabilities Conference (CSUN)** (March 2013) - Jim Tobias (Inclusive Technologies), Rich Schwerdtfeger (IBM), and Madeleine Rothberg (NCAM) presented on GPII and the specific objectives of the PGA project to a group of assistive technology users and providers. The closing discussion allowed the team to collect feedback from the audience on what tools they use to collect preferences, as well as their insights on design and feature considerations for preference collecting tools, and factors supporting user customization.
- **Open Courseware Consortium Global Conference** (Indonesia, May 2013): Jutta Treviranus engaged an international community of educators in providing input regarding preference management in an accessibility workshop at the global OCWC conference in Indonesia.
- **Designing Enabling Economies and Policies Conference (DEEP)** (July 2013): Jutta Treviranus convened an international community of stakeholders who were engaged in providing input to PGA. This included numerous end-users and potential implementing organizations.
- **North Carolina Rehabilitation Association/North Carolina Assistive Technology Program Conference** (August 2013) – Jim Tobias (Inclusive Technologies) led a webinar entitled, *Global Public Inclusive Infrastructure (GPII) Webinar: Building Accessibility into the Cloud*. The goal of the webinar was to present GPII in terms of how it works and how it potentially fits into how practitioners do their work, as well as PGA-related content including a discussion of user control over their technology through preference gathering tools, and roles of practitioners in supporting preference gathering processes. The 35 webinar participants spanned state policymakers, staff members of state assistive technology (AT) programs, AT practitioners, special educators, rehabilitation counselors, and occupational and speech therapists. The webinar discussion led to increased awareness on behalf of some participants around what they deemed as much needed, centralized tools to support practitioners' work in evaluating clients, as well as increased overall awareness of new technologies that move beyond the field's current use of familiar AT products and practices that are potentially overly clinical and outdated.
- **Rehabilitation Engineering Research Center for the Advancement of Cognitive Technologies Conference (RERC-ACT)** (October 2013): Jim Tobias (Inclusive Technologies) presented on inclusive employment and how cloud-based systems can provide workplace

support. The presentation also addressed components of the PGA work, and specifically the project's identification of features for people with cognitive disabilities.

- **Internet Governance Forum Conference** (October 2013): Jim Tobias presented on Rights Issues for Disadvantaged Groups in International Internet Policy and how GPII will collect user needs and preferences via PGA products
- **International Technology and Persons with Disabilities Conference (CSUN)** (March 2014). Jim Tobias (Inclusive Technologies) and Jess Mitchell (OCAD) are currently preparing a proposal to present on GPII and the outcomes of the first phase of the PGA work.

Meetings and Workshops

- **EpolicyWorks Cross-pollination Meeting** (January 2013): Lisa McLaughlin (ISKME) participated in this meeting organized by Clayton Lewis, which sought to bridge connections across accessibility-focused, federally-funded initiatives. Participation in this meeting helped to raise awareness of the PGA project across partners from several projects including: An accessibility EAG grant working to look at computerized assessments for students, a voting project focused on developing usability and accessibility standards for voting, an SRI initiative looking to develop accessibility standards for the Learning Registry in collaboration with Benetech's A11y Metadata initiative, and the Dynamic Learning Maps project, among others. This meeting led to the inclusion of several projects in the research deliverable inventory of existing initiatives, as well as deepened connections with others doing work related to PGA.
- **ISKME Metadata Workshop** (January 2013): ISKME held a webinar with PGA team members and listserv participants (including organizations like NSTIC) to build knowledge about potential roles of metadata in the profile creation process. The workshop led to new partnerships between the PGA community and the data standards communities, including leading to ISKME's participation as a test platform for the A11y Metadata initiative.
- **Learning Resource Metadata Initiative Workshop** (September 2013): Lisa McLaughlin (ISKME) participated in this workshop of 30 attendees invested in initiatives focused on improving how digital learning resources are described for discoverability on the Web. The attendees included team leads from the Federal Learning Registry project and the Learning Resource Metadata Initiative. The workshop addressed the need to embed accessibility standards into larger initiatives such as LRMI to ensure that tagging resources for users with diverse needs is seen as a central rather than a supplemental function. There was active debate about where approaches for tagging and identifying learning resources that support accessibility were deemed to have too high of a time cost. Lisa offered several examples from the PGA team's cognitive research indicating that more rather than less attention to detail would be needed to facilitate the creation of content that could be served to profiles. In doing so, the workshop helped to increase awareness among the attendees that granular metadata is needed for discovery of accessible learning resources and the adaptive learning engines envisioned for 21st century learning. New efforts developed as part of the convening to establish a governance structure to evaluate overlap between new extensions emerging around LRMI including the accessibility standard that will provide productive links between these initiatives.

- **Cloud and Accessibility Workshop** (NSF Offices, June 2013): During this NITDR-sponsored consultation Gregg Vanderheiden and Jutta Treviranus presented the findings of PGA and requested input to the deliberations from the group of assembled experts.
- **MEnabling Summit** (June 2013): Jutta Treviranus presented PGA results and design proposals and sought input from a broad international audience of mobile developers, service providers and end users in Washington.
- **United Nations Expert General Meeting** (New York, July 2013): As an invited expert to the UN Assembly consultations on post- 2015 goals and the Convention on the Rights of People with Disabilities), Jutta Treviranus presented the potential of PGA and consulted with international delegates for input on required preference domains.
- **ISO JTC1 Meetings** (Moscow, Sept. 2013): Jutta Treviranus presented the results and advances of the PGA project to the international working group and ISO Plenary in Moscow. Recommendations for additions and revisions to the standard coming from PGA research were presented at the meetings.
- **Google Accessibility Summit** (September 2013): Gregg Vanderheiden and Jutta Treviranus presented the on the findings of PGA to Google staff and the international advisory panel gathered by Google.
- **International Assistive Technology Summit** (Sao Paulo, October 2013): Jutta Treviranus presented the findings and recruited the input of global assistive technology researchers at this global summit.
- **Financial Inclusion Global Summit** (London, October 2013): Jutta Treviranus will be gathering preference considerations related to financial inclusion at this international summit.

Field Expert Conversations

- **Keith Hazelton, Sr. IT Architect, UW-Madison** (January 2013): Lisa McLaughlin (ISKME) and Keith met to discuss synergies between the PGA work and the National Strategy for Trusted Identities in Cyberspace. The conversation led to Keith's deepening involvement in the Cloud 4 All privacy and security work. Keith also attended project webinars such as ISKME's metadata workshop, and joined the PGA listerv, where he initiated discussions around privacy of user profile data.
- **Heather Hartkopp, Project Manager, Dynamic Learning Maps & Cloud Computing** (April 2013): Cynthia Jimes (ISKME) and Heather Hartkopp met to discuss synergies between PGA work and the Dynamic Learning Maps Project. The discussion provided insights on the ways that preference management tools can leverage existing data to populate user profiles, and brought to light findings from her team's survey with special education teachers to assess needs and preferences of students with disabilities.
- **Lisa Seeman, Sr. Consultant, Deque Systems, founder of UCB Access and Athena ICT** (April 2013): Richard Schwerdtfедder (IBM), Lisa Petrides (ISKME), and Cynthia Jimes (ISKME) met with Lisa Seeman at the ISKME office to discuss the PGA project. Discussion of the PGA project's work and challenges during Lisa Seeman's visit allowed for a rich discussion driven in part by her questions around strategies for capturing and matching preferences and needs, and cognitive

preferences in particular. Lisa joined the PGA listserv as a result of the meeting, and provided feedback on an early draft of the research deliverable for the project.

- **Feedback-gathering Conversations with Individuals with Disabilities** (July 2013): Shari Trewin met with three contacts at the United Cerebral Palsy Center to collect early input on the preference categories and preference names that had been developed through the PGA work, as well as the individuals' experiences with preference setting on their current devices.

Social Media and the Web

- **Listservs** (ongoing): A "global preferences" listserv was established early in the project as a mechanism for knowledge sharing and debate around project findings, insights, questions, and logistics. Posts to this list by the PGA team were, where relevant, also posted to the Cloud4All and Fluid lists, as well as the World Wide Web Consortium list, as a mechanism to solicit feedback and input from affiliated communities. Field experts were, on an ongoing basis, also invited to join and contribute to the PGA project listserv.
- **Website Dissemination** (August 2013): The PGA research deliverable was posted on both the World Wide Web Consortium website and the ISKME website. A press release about the report was also disseminated by ISKME through prlog.org.
- **Twitter and Facebook** (ongoing): Several SMEs and PGA team members, including Richard Schwerdtfeger (IBM), as well as Lisa Petrides, Lisa McLaughlin, and Cynthia James (ISKME) have generated tweets and Facebook posts on the PGA work, learnings, and deliverables.

Other

- **New World-Wide-Web Consortium Cognitive Accessibility Task Force** (ongoing): With the research deliverable we have the individual adaptations required to satisfy the broad range of cognitive impairments. By coupling these requirements with the conveyance of user preference we now have the seeds to build a road map in the World Wide Web Consortium to address cognitive access as part of a new Cognitive Accessibility Task Force. The value of the road map, as we learned with WAI-ARIA, is that we can use it to show where the gaps in accessibility needs and define a strategy to address them that can be viewed and reviewed by the entire web community. The advance of mobile devices as a platform will help accelerate that effort as all users will constantly be in a state of cognitive impairment and will benefit from what we have learned from PGA Research Deliverable 1. Rich Schwerdtfeger will take an active role in moving what we learned from this project into the development of the cognitive roadmap in the task force. The task force will work directly with the Web Accessibility Initiative Coordination Group to execute on the strategy.

Appendix C: Project Background Information

Relationship Between PGA and GPII

In this document we refer to the PGA Preference Tool Framework/Architecture and the GPII Preference Framework and the GPII “auto-personalization from preferences” (APfP) Architecture as different items. In fact, the PGA Preference Tool Framework is part of the overall GPII Preferences Framework which is in turn part of the overall GPII “auto-personalization from preferences” (APfP) Architecture. All of the work here is closely coordinated with the work in the other projects (Cloud4all, Fluid/Floe, Prosperity4All, Trace, etc.) working on development of the GPII. In this document we refer to the PGA Ecosystem and Framework/Architecture to identify the work being carried out in this project. Although other projects have some work on preferences it has been constrained to that required to support their core goals and work. For example, in Cloud4all the primary focus is on “auto-personalization from preferences” (APfP) capability and not streamlined preference acquisition designed to support a broad range of users. There is some preference tool work – but it was included only because it was needed to create Needs & Preferences sets for testing the “auto-personalization from preferences” (APfP) capabilities of its prototypes. The PGA project has become the focus of all of the preference tool work and is now the coordinating project for all of this work. The work, definitions and architecture for preference tools is all coming out of this project (working with the other project teams) – and is the work and architecture being adopted by the other GPII projects.

References

- Fisk, A., Rogers, W. Charness, N., Czaja, S. and Sharit, J. (2009). *Designing For Older Adults: Principals And Creative Human Factors Approaches*. Boca Raton: CRC Press.
- Forrester Research (2004). "Accessible technology in computing—examining awareness, use, and future potential." Online Forrester research report. Retrieved from <http://download.microsoft.com/download/0/1/f/01f506eb-2d1e-42a6-bc7b-1f33d25fd40f/ResearchReport-Phase2.doc>
- Fox, S. (2011). "Americans living with disability and their technology profile." Online Pew Research Center report. Retrieved from <http://pewinternet.org/Reports/2011/Disability.aspx>
- Gajos, K., Wobbrock, J., and Weld, D. (2007). "Automatically generating user interfaces adapted to users' motor and vision capabilities." In *UIST '07: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, pp. 231-240. New York, New York: ACM.
- Hastings Kraskowsky, L. and Finlayson, M. (2001). "Factors affecting older adults' use of adaptive equipment: Review of the literature." *American Journal of Occupational Therapy*, 55(3), pp. 303-310.
- Hudson, R., Weakley, R., and Firminger, P. (2004). *An Accessibility Frontier: Cognitive Disabilities And Learning Difficulties*, Retrieved from <http://usability.com.au/2004/12/an-accessibility-frontier-cognitive-disabilities-and-learning-difficulties-2004/http://usability.com.au/2004/12/an-accessibility-frontier-cognitive-disabilities-and-learning-difficulties-2004/>
- Iyengar, S. and Lepper, M. (2000, December). "When choice is demotivating: Can one desire too much of a good thing?" *Journal of Personality and Social Psychology*, 79 (6), pp. 995-1006.
- Lenker, J., Harris, F., Taugher, M. and Smith, R. (2013). "Consumer perspectives on assistive technology outcomes." *Disability and Rehabilitation: Assistive Technology*, pp. 1-8.
- Lesh, G.W., Rinkus, G.J., Moulton, B.J., & Higginbotham, D.J. (2000). "Logging and analysis of augmentative communication." In *proceedings of the RESNA 2000 Annual Conference*, pp. 82-85. Arlington, VA: RESNA Press.
- Mann, WC., Hurren, D., and Tomita, M.R. (1993). "Comparison of assistive device use and needs of home-based older persons with different impairments." *American Journal of Occupational Therapy*, 47(11), pp. 980-987.
- Nicolau, H. and Jorge, J. (2012). "Elderly text-entry performance on touchscreens." In *ASSETS '12: Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, pp. 127-134. New York, New York: ACM.

- Reimer-Weiss, M.L., & Wacker, R.R. (2000). "Factors associated with assistive technology discontinuance among individuals with disabilities." *Journal of Rehabilitation*, 66(3), pp. 44-50.
- Scherer, M. et al., (2005). "Predictors of assistive technology use: the importance of personal and psychosocial factors." *Disability and Rehabilitation*, 27 (21), pp.1321-1331.
- Spool, J. (2011). "Do users change their settings?" Online article. Retrieved from <http://www.uie.com/brainsparks/2011/09/14/do-users-change-their-settings/>.
- Trewin, S. (2004). "Automating accessibility: the dynamic keyboard." In ASSETS '04: Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility, Atlanta, Georgia, pp. 71-78. New York, New York: ACM.
- Zickuhr, K. (2013). "Who's not online and why." Online Pew Research Center report. Retrieved from <http://www.pewinternet.org/Reports/2013/Non-internet-users.aspx>.