PixieEngine: Context-sensitive Interface

Deng-Shun Chang, Andrew Grossman, Michael Smith, Jacob Wells
MID I, University of the Arts
Dr. Slavko Milekic, Interactivity Studio
May 7, 2010

PixieEngine is a new mobile technology developed by Human Network Labs (HNL) that allows users access to information from their physical surroundings relative to their location. As this technology is still in development its full capabilities have not been reached, though we have good understanding of what it will be capable of and what the future of the PixieEngine could be.

PixieEngine is an extremely smart and powerful technology. It physically takes the shape of a computer chip and can connect to other mobile devices via Bluetooth. PixieEngine enabled devices can communicate directly with other PixieEngines without requiring supportive infrastructure. This is a huge strength of the technology especially when compared to existing mobile technologies that rely completely on vast amounts of infrastructure for connectivity between users and devices. They communicate through a 2.4Ghz radio frequency (FCC Licensed) with a data transfer bandwidth of 1 Mbps. Comparatively, current mobile technologies using 3G technology only have 384 Kbps of data transfer bandwidth.

Over this radio frequency PixieEngines can track each other in real-time with an update rate of 0.1 seconds. They have an accuracy of plus or minus one meter when tracking and locating other devices. Tracking capabilities not only include general location, but also orientation and elevation. PixieEngine can determine your path/direction of movement and can orient other devices around you relative to yourself. This means that you can see which direction other users are facing and even what floor of a building they are located on. Currently the technology has a range of a 50m diameter indoor and a 200m diameter outdoor. The range is relative to the user's position; as the user moves the range moves with him. As a visual reference you could consider a bubble around the user that represents his range of connectivity. Within this range, the user can connect with up to 200 other devices. This includes user-operated devices as well as spotcasts.

Spotcasts are stationary PixieEngines that have Ethernet connectivity and can project information on stationary objects around them creating smart objects. These devices could be particularly useful in retail operations. For example, a grocery store could use spotcasts as a way to keep track of items on a shelf and at the same time provide useful information to the customer such as nutritional information or price. Spotcasts could also be used for advertising. By using a spotcast's ability to detect users as they approach, advertising could potentially be catered to the individual's navigation history. Another capability of spotcasts is that they can record a user’s navigation by connecting with the user's device and sending information to a main hub through an Ethernet connection. This creates a history of navigation for each individual user and can be used to provide for the user’s specific needs/desires or, as we will discuss later in this paper, can aid in detecting a change of context and can provide a user with information that is most relevant to him.

Currently the PixieEngine has the physical dimensions of 45mm x 55mm x 4mm with casing and rechargeable battery. Data storage is variable, based on the size of the microSD card used in the device. Plans for the PixieEngine’s debut are limited to asset tracking using a single user-operated device with a limited interface. The asset-tracking device will be linked to items such as a purse, backpack, vehicle or even a child. Items will be fashioned with non-interfaced PixieEngines, as they able to be kept track of by a single user-operated device.

Although this is a starting point for the development of this technology, we foresee the bright future of PixieEngine and what it can truly offer as a powerful mobile device. The scope of our research and design looks three to five years out. The quality of PixieEngine is a perfect match for a mobile device that can track the motion of everything and everyone and further provide detailed information in users' immediate surroundings. With this understanding of PixieEngine's capabilities, our design process follows.

Our design challenge consists of two parts. Firstly, since PixieEngine is a newly developed technology and there is no actual product existing in the market, people have no idea what PixieEngine is. Therefore, in order to let people understand our design, we have to introduce the technology to them. Instead of listing the technical specs, we chose to illustrate what PixieEngine is and what it is capable of. With these simple illustrations, we hope to reach not only the hardware and software developers but also the potential users in the general public.

Secondly and more central to our focus, our challenge concerns utilizing the strength of PixieEngine to provide users with better information then other mobile devices do. In this forever changing and information-overloading world, people are constantly bombarded with different kinds of information through various media. Current mobile devices are capable of assisting users to navigate around their environment using a combination of cellular technology, wireless Internet and GPS technology. However, there is not yet a device that can track real-time motion of people and transportation while providing sufficient information at the same time.

With the advance of technology, more and more detailed information can be provided to users on the go. Therefore, with the size constraint of a mobile device in mind, how can a large amount of information be simply displayed and easily navigated by a PixieEngine user? Our approach to this challenge is not only focusing on the visual display of PixieEngine, but also, focusing on sorting and dissecting different types of information. We then further prioritize information with a logical standard. The stakeholder of our design is the developer of PixieEngine, HNL and other potential hardware/software developers who would like to explore

PixieEngine more deeply and further improve this technology. Most importantly, we would like to present our design to current mobile device users who want to know more about their surroundings.

To approach this problem and reduce visual clutter on the PixieEngine-powered mobile device, we designed a software interface that will display only the information that is most relevant to the user. To accomplish this, we began by first establishing a visual language for the PixieEngine display that distinguishes relevant information from irrelevant information. We then developed a relevance bias that maintains an information hierarchy by prioritizing information according to changing contexts. We also designed an interface permitting the user to navigate through multiple layers of available information.

When considering the different types of information that could be obtained through PixieEngine, we found that all of the information could be broken down into three elemental groups: people, spotcasts (places), and transportation We used different shapes to indicate the different types of information on the mobile display; circles were used for people, triangles for spotcasts/places, and squares for transportation.

Since these three groups of information could be broken down into subgroups, we chose different colors to distinguish between subgroups in each group. Based on NASA standards, we assigned a color to each subgroup. For the circle-shaped people markers, we chose blue to represent “friends” and red to represent “unknowns”. The spotcasts group required the most organization because there are so many types of places to distinguish between. When mapping out all the different types of places and considering the cognitive load required for the user to correctly associate types of data to many colors on a mobile display, we strived to lessen the amount of subgroups to as few as possible. We ended up breaking the spotcasts group into five subgroups and colors: municipal (navy blue), retail (green), dining (pink), recreation (purple), and utility (orange). Although the last group, transportation, could be broken into subgroups of modes of transportation, we decided not to divide it into subgroups with different colors because we did not want to increase the cognitive load by introducing more colors when the modes of transportation could be easily distinguished using well-known icons.

After identifying the subgroups with different colors, the subgroups were then broken down into types of places by applying iconography to the visual markers. We assembled a sample set of icons based on icons developed by Google, AIGA, and other designers. Because these icons are recognizable to most mobile device and web users, they serve as scaffolding to our visual language until the user becomes comfortable associating different subgroups and groups with the correct colors and shapes. The set of icons we used is in no way meant to be comprehensive; more icons will surely be utilized at a later development stage. We also considered that many businesses would want to display their custom icons, just as photographs may be useful to distinguish between people on a mobile display. However, using custom icons and photographs as visual markers on a mobile display is most useful when there is a small number of markers on the screen and the markers are large enough on the screen to show adequate detail within. Considering this, we decided that custom icons and photographs may be used at smaller scales (e.g., when the user is in a small coffee shop) versus simple icons, colors, and shapes used at larger scales (e.g., when the user is on a busy city street).

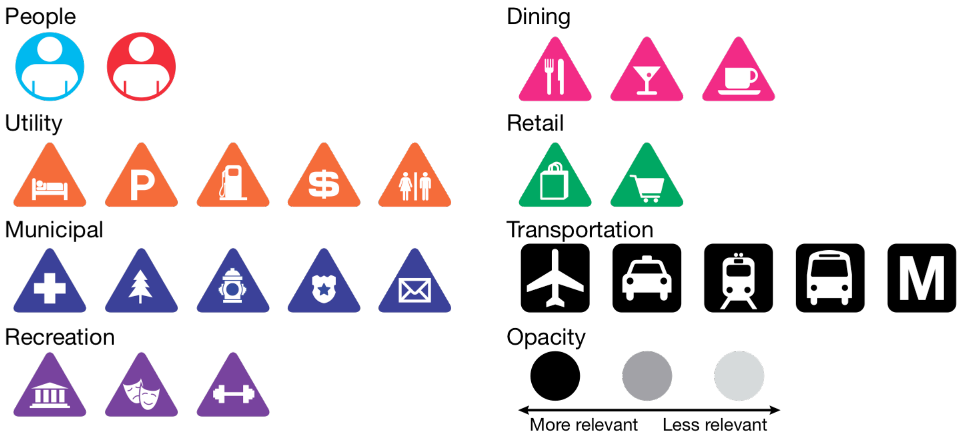


Figure 1. Iconography

Opacity was chosen to represent how relevant each marker is at the current time. The more opaque a marker is, the more relevant it is to the user. We decided to use only three levels of opacity to minimize the amount of cognitive load required to distinguish between each level while still allowing PixieEngine to provide sufficiently detailed information about relevance.

Using the affordances of PixieEngine technology, a PixieEngine can determine what contexts the user is engaged in. PixieEngine is range-aware to a much greater accuracy than other technologies currently used in mobile devices. It can also communicate with other PixieEngines to receive and send data in nearly real-time. These affordances allow PixieEngine to process enough data about the user's surroundings that it can predict and respond to exactly what context the user finds himself in far more accurately than current mobile devices.

Through researching the different contexts involved in location-based services, we identified the following six contexts that could be most used by the PixieEngine relevance bias (Nivala and Sarjakoski; Stefan, Neun, and Edwardes):

User Identity: Traits of the user such as age and gender, personal preferences, who their friends and colleagues are.

Location: Absolute or relative location. PixieEngine can only use relative location data on its own. However, it can piggyback onto other technologies such as GPS to use absolute location data.

Time: Time of day, morning, afternoon or evening, day of the week, month, season of the year, etc.

Purpose of use: Activities, goals, tasks and roles of the users. This is difficult for most technologies to predict accurately; generally, this should be indicated by user input.

Social and cultural situation: Indicated by user's proximity to others, relationships to others and collaborative tasks

Navigation history: Where the user has been and what the user has done.

We developed a logic system to prioritize all available information based on these contexts. Using information gathered by PixieEngine, the user's context can be determined and the display can adapt accordingly. The rules that govern how the display markers adapt to changing contexts are laid out at the end of this paper (see attached chart, "Relevance Logic Sample"). This relevance bias logic affects a default level of relevance attributed to each icon type as the user experiences changes in contexts. This default level of relevance can be customized according to the user's preferences. Examples of how the PixieEngine display adapts to contexts using this relevance bias system are shown in Figure 2.

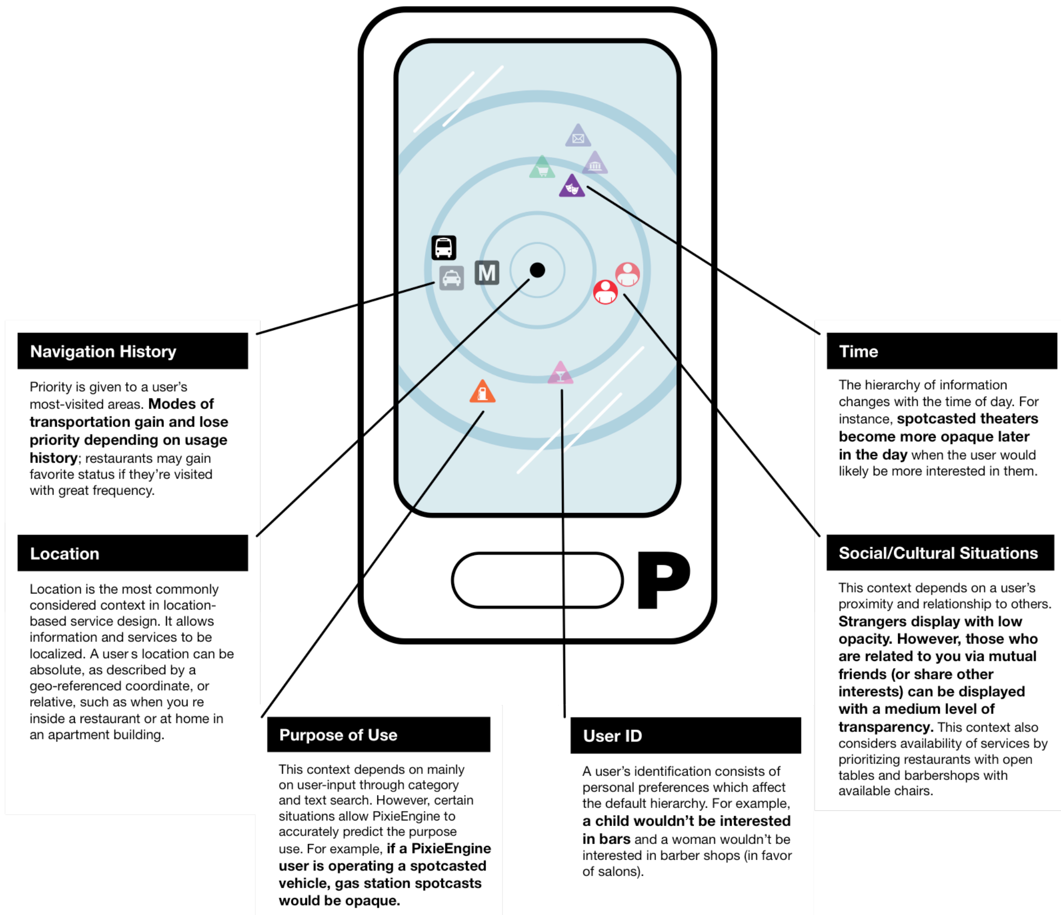
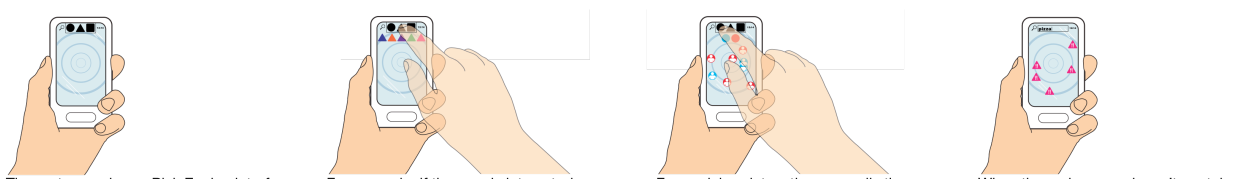


Figure 2. Adaptive display based on changing contexts

Ideally, the relevance bias will provide the user with the most relevant information without their having to manually search, but we are aware that the relevance bias will fail to deliver the most relevant information to the user in some cases. Because of this fact, we included the options of manual search functions that provide the user with absolute control over the PixieEngine display. This allows our design to gracefully degrade when the relevance bias is incorrect and function appropriately when it is correct.

To allow the user to manually navigate through the levels of available information, we developed a simple and efficient interface. The root menu consists of four interactive elements: the text search feature, people locator, places locator, and transportation locator (see Figure 3).

When the user selects one of these locator options, the display reveals the colored shapes of that group’s subgroups. The user can then choose a subgroup and all of the display markers within that subgroup will be displayed at full opacity, allowing the user to examine all available options, regardless of the level of relevance assigned to the markers by the relevance bias. The text search can also be used to find nearby people, places, and transportation options of interest. This works in the same way as the locator functions; all markers associated with the text query will be displayed at full opacity on the PixieEngine display.



The root menu in our PixieEngine interface consists of four main interactive elements: the search feature, people locator, places locator and transportation locator.

For example, if the user is interested in places to eat, he or she calls up the “places” menu option.

For social updates, the user calls the “people” menu option.

When the main menu doesn’t contain precisely what the user is interested in, a search box can be used to quickly narrow results.

Figure 3. Interface considerations

These features and technologies are not individually new; online services including Google Mobile, Yelp, Loopt, TellMeWhere and others perform a range of social and locative services which mimic PixieEngine’s. Looking at Google Mobile, for example, we find capabilities such as mapping, plotting friends’ locations, opinions, reviews and public transportation timetables. Although these features are available on a user’s mobile device (via Maps, Buzz, etc.) they each depend on access to cellular or Wi-Fi infrastructure to fetch their data. This data must be fetched on an as-needed basis and is also unable to account for the 63 bus running late or the Pizza parlour having closed last week (both things PixieEngine can learn and display on-the-fly via Spotcasts).

A service such as Yelp likewise depends on the user to request specific information about a location or category, with some amount of prior knowledge. The results are then only visible if the device is connected to a wide-area network and resolve detail only as fine as the integrated mapping software allows. Consequently, the Yelp user must pair these results with familiarity of the area or GPS hardware and additional software. Your content, by default, is visible to all who visit Yelp, not only your friends. The PixieEngine user, on the other hand, may simply orient himself according to the spotcasts on-screen.

Loopt, whose tagline is "discover the world around you," might seem like a PixieEngine competitor on the surface. A closer examination reveals that the service depends on your manually "checking-in" to locations if you want them to be broadcasted to your networked community of friends. The company provides comprehensive listings, which you can comment on and share, although this solution depends on wide-area network infrastructure similar to Google Mobile and Yelp and is glaringly context-insensitive. In other words, the user/device equipped with this service must provide input at each destination for others to search. In late 2007, Yelp introduced facebook integration, allowing registered users to post reviews and status updates from one service across the other (a model now popular with nearly all online social services). Contrary to this model, the PixieEngine functions based on the user's active profile eliminating the need to remember to "check-in" throughout the day – a great reduction in cognitive load.

TellMeWhere works on the "check-in" model as well, however it touts a strong connection to your local community and network of friends. Based on your recommendations in addition to those your friends post, the service suggests new destinations you're inclined to enjoy. Like the three services above, this too depends on manual input to generate content required for their server-side guesswork. Ideally, and in the case of the PixieEngine, your device detects patterns in your travels and destinations. These patterns then become learned preferences, such as a predisposition to catch the train before a bus or to grab a coffee before arriving at the office. These services require that you manually search for these things for them to be displayed on your device. PixieEngine, at your discretion, allows you to see these things automatically and with great detail, using only available contextual information. And while these services all have a wide area of scope -- ranging from street-level views to entire countries and everything in-between -- they don't allow you to see friends, acquaintances (and others) moving about in near-real-time (0.1s refresh) and with accuracy down to plus-or-minus one meter as PixieEngine does.

Our ideas for further development include a more robust visual language, improved relevance bias and a material prototype of our visual interface proposals. The improved visual language will ideally include a far more detailed view of the complex relationships between people while maintaining the graphic simplicity of our initial designs. Additionally, improved and expanded iconography will include detailed markers for "valuable assets", thereby taking advantage of the high-resolution accuracy afforded by the PixieEngine technology. Our next research phases should include a greater focus on the small-scale (as opposed to the macro view familiar to us from current social network services). And while all these future steps require that we write context sensitive software to be used for testing, the first prototypes can be demonstrated similarly to how we presented the early steps: using animation and visual diagrams.

Relevance Logic Sample

	Icons	Default	Location		Cultural and Social Situation			Time	Navigation History		User Identity	Purpose of Use
			"Favorite" Option	Indoor vs. Outdoor Space	High Density (>40 People)	Low Density (<15 People)	Related & Unknown	Not During Operation Hours	Most visited* areas/Most used modes of transportation	Ignored** places, modes of transportation	Age	Near car PixieEngine
People	Friend	1		If not in space, 2								
	Unknowns	3		If not in space, 3	**3**	2	If related, 2					
Transportation	Bus	2		If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Subway	2		If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Train	2		If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Airport	2		If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Taxi	2		If not in space, 3				**3**	Default = Default -1	Default = Default +1		
Utility	Hotel	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Parking	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		1
	Restroom	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Gas	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		1
	Bank/ATM	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
Dining	Food	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Bar	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1	If younger, 3	
	Coffee	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
Retail	Shopping	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Grocery	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
Entertainment	Gym	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Theater	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Museum	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
Municiple	Hospital	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Park	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Fire Dept.	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Police Dept.	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		
	Post Office	2	**Default = 1**	If not in space, 3				**3**	Default = Default -1	Default = Default +1		

Levels of relevance: 1 = , 2 = , 3 =

Bold = trumps other setting

*Visited = PixieEngine remained in spotcast's "shadow" for a certain period of time (e.g., 20 minutes, two times per month).

**Ignored = PixieEngine did not "visit" a familiar area for a certain period of time (e.g., one month)

This chart represents the basic viewing mode, not the search mode. It considers only this sample of icons and is not meant to be comprehensive. Other contexts may apply to future points of interest.

Citations:

Mogford, Richard H.. "Color Usage Site". NASA. April 7, 2010 <http://colorusage.arc.nasa.gov/>.

Nivala, Annu-Maaria, and Tiina L. Sarjakoski, "An Approach to Intelligent Maps: Context Awareness." *Workshop "HCI in mobile Guides"*. Udine, Italy: 2003. Print.

Steiniger, Stefan, Moritz Neun, and Alistair Edwardes. *Foundations of Location Based Services*. 1.0. Zurich, Switzerland: University of Switzerland. Print.

"Symbol Signs." *AIGA - The Professional Association for Design*. AIGA, 2010. Web. 7 April 2010. <http://www.aiga.org/content.cfm/symbol-signs>.