Personal Rover User Experience

Emergent Design, Inc.

Tuesday, January 15, 2002

e m

e r g

e n t/d

e s i g n

Notes:

Table of Contents

Personal Rover User Experience

This document describes elements of the user experience for interacting with a Personal Rover[*]. This document is intended to aid in the conceptualization of the Personal Rover product. Specifically, this document is designed to seed the prototyping process of the Personal Rover User Interface.

The document is organized as follows:

- **Goals and Objectives** - This section defines the higher level goals and objectives for the Personal Rover. These goals and objectives relate the capabilities of the Personal Rover to useful and instructive roles for the toy as both a product, and as a instructional and promotional vehicle of appeal to sponsors of the product and affiliated programs.
- **Definitions and Assumptions** - This section defines the baseline assumptions about the Personal Rovers core product features and functions. These definitions and assumptions form the basic building blocks available for the orchestration of more involved user experience scenarios.
- **Principles of Operation** - This section defines the Personal Rover's modes of operation, programming, and control from the user experience standpoint. These principles provide a basic vocabulary of operations, means of control, and things to do with the robot outside of the formal structure of a (tightly or loosely) structured mission. Beyond the basic product feature definitions of the previous section, these principles define various useful modes and skill sets that the product supports and helps users master.
- **Personal Rover Missions and Scenarios** - This section outlines a set of prototypical missions and scenarios that can be realized from the toy's basic capabilities. These mission scenarios are not exhaustive, but serve to illustrate a set of basic 'use cases' that give a representative overview of the Personal Rover's usefulness as a successful toy product that exhibits strong synergy with the goals of robot science and exploration of appeal to the product's sponsors. This section will present a series of believable stories that relate the capabilities of the

[*] The term 'Personal Rover' is used throughout this document as a placeholder for the product name.

Personal Rover to the experience of those programming and using the toy.

- **User Interface Requirements Summary** - This section consolidates the user interface implications suggested by the descriptions in the document. This set of interface implications and functional use cases serves as a checklist for conceptualization of supporting user interface elements by the UI Designer.
- **Next Steps** - This section summarizes this report and offers suggestion of how this information can help in refinement of the product definition, user experience, and user interface prototyping efforts going forward.
-

Important Note: This document is intended as a precursor to the development of user interface prototypes and designs. This document is not intended as a user interface design document. The descriptions of user experience will necessarily make suggestions about user interface components as concepts. However, the concrete realization of the interface tasks and elements outlined in the experience is the role of the user interface designer and UI prototyping process.

Goals and Objectives

This section defines the higher level goals and objectives for the Personal Rover. These goals and objectives relate the capabilities of the Personal Rover to useful and instructive roles for the toy as both a product, and as a instructional and promotional vehicle of appeal to sponsors of the product and affiliated programs. These goals and objectives will be used as guiding criteria for defining the Personal Rover user experience.

TARGET AUDIENCE

The user experience for the Personal Rover will address the following primary audience and objectives:

Children (male and female), 9 to 13 years of age. The target audience will assume to also possess the required computer and communications infrastructure necessary to program the Personal Rover.

Note: Due to the programmable capabilities of the Personal Rover product platform, the appeal of the product and supporting experience will extend beyond the core target demographic. Ideally, the inherent extensibility of the product and its supporting community will extend to teens and adults able to use the lower-level and more advanced features of the product platform to develop missions and productive applications of the Personal Rover beyond those defined by the core off-the-shelf product. While the off-the-shelf user experience will necessarily cater to the target demographic, the Personal Rover's inherent programmability and extensibility will permit development and insertion of user interfaces and experiences that target and evolve to all audiences.

DEFINING THE PRODUCTIVE HOME PERSONAL ROVER CATEGORY

The Personal Rover product and its supporting programming environment and user experience will promote awareness and utility of personal in-home robots as useful adjuncts to the child's daily life. The collective experience of owning, programming, and relying upon the facilities of the Personal Rover will define productive and appealing uses for the Home Personal Rover as a valuable and viable product category for its initial target audience. The Personal Rover user experience and user interfaces will be designed to demonstrate, facilitate, and encourage productive employment of the robot within this new category.

The utility to the core target audience will emphasize learning about robots, automation, and programming as essential skills in today's world. The Personal Rover experience will also teach productivity programming and the relationship between people and humane and productive control of intelligent agents and their wireless and remote control mechanisms. Successful exposure to productive robotic applications to the target audience will seed further penetration of home and Personal Rovers and agents as the target audiences matures in automation sophistication and spending power.

> **Important Note**: While the product may support characteristics and modes of play similar to 'virtual pet' products, the Personal Rover product and its user experience will be purposefully engineered to highlight and extend the practical and useful roles of a Personal Rover toward science-oriented education and practical productivity tasks. The user experience will highlight and emphasize such productive roles for the toy, providing a value proposition that positions the product as distinct from, above and beyond its 'virtual pet' precursors.

A UNIQUE EDUCATIONAL CHANNEL AND MOTIVATOR

The affinity of the Personal Rover with national science and education programs and objectives, especially NASA's upcoming Mars explorations, will provide another rich market center. As a learning tool the Personal Rover will support the following:

- Provide a motivational learning platform for robotics, computer vision, programming, scientific method, physical, planetary, and life sciences.
- Provide a platform that encourages both the user's experimentation as well as the ability for the robot to act as a teaching assistant to draw out and challenge the user along both structured and unstructured curricula.
- Provide a adaptive, personalized interface and evolving relationship between the robot and the owner in order to keep the toy fresh and interesting over time and over different types of studies and endeavors.
- Support both structured (mission-oriented) and unstructured (free play) modes of engagement with the toy.
- Increase affinity for and understanding of NASA's planetary exploration missions through integration of home robot missions with Mars missions, science and telemetry. Such experiences will increase appreciation for the role of autonomous robots in those missions, and the robot-human interface and joint work required in such missions. Similar promotional relationships could be developed for other product sponsors.

DEVELOPMENT OF A RICH, SELF-SUSTAINING USER COMMUNITY

The Personal Rover will form a natural center of attraction for an active user community. The user community will help encourage the product's success through both product sponsor's and organically emerging online and real-world forums. The Personal Rover's encouragement and use of open-source initiatives will serve to seed and encourage community sharing and participation.

Consider the following example roles of both online and real-world community support and sharing opportunities:

- Publication of open standards for development and product extensions.
- Forums for the uploading and downloading of new product programming software.
- Forums for the uploading and downloading of new missions and curricula, including supporting software.
- Forums for the sharing of experimental results and rich media that demonstrate Personal Rover's and the users in action.
- Active forums for social and informational exchange around the Personal Rover, including chat, message boards, newsgroups, instant messaging, etc.
- Staging of Personal Rover events, such as at museums, science fairs, schools, children's social clubs and organizations, etc.
- Staging of Personal Rover challenges and competitions that center competitors around structured tasks, races, logic and programming challenges, etc.
- Cooperative linkage between Personal Rover communities and dedicated science and learning sponsors, such as NASA, that highlight and encourage analogs to real-world robotics tasks and

operations with the Personal Rover through exchange of telemetry data and experimental results.

Definitions and Assumptions

This section defines the baseline assumptions about the Personal Rover's core product features and functions. This includes a preliminary definition of the following baseline Personal Rover product architecture.

PRODUCT ARCHITECTURE

This user experience guide makes the following assumptions about the Personal Rover product architecture.

- A baseline Personal Rover (see details below).
- The potential for add-on and 3rd party enhancements to the baseline Personal Rover to include things like additional end-effectors, sensors, environmental markers, etc. While these yet-to-be-defined enhancements are not able to be addressed in this user experience guide, the intent is for the user experience to be readily amenable to incorporation of such enhancements (See also Web-enabled support centers below).
- A standardized Programming and Control Center. This will include at least one computer-based user interface that supports the basic programming, mission control, tele-operations, and educational curricula that form the extensible core of the off-the-shelf product.
- Web-enabled support centers. These support centers may include both Personal Rover sanctioned sites, as well as 3rd party sites and designated open-source initiatives and support for augmenting the Personal Rover's programmable functionality and curricula through sharing within an active online user community.
-

BASELINE PERSONAL ROVER CAPABILITIES

The following describes a set of baseline Personal Rover features and functional elements. These definitions and assumptions form the basic building blocks available for the orchestration of more involved user experience scenarios.

For purposes of proposing possible user experience scenarios this document will assume the Personal Rover has the following baseline capabilities:

- The ability for the Personal Rover to measure distances or distance estimates based on motion (readings from stepper motors?).
- The ability of the Personal Rover to triangulate its position from measurements taken from two or more distinctive reference markers.
- The Personal Rover will have the capability to execute pre-defined user programs. These programs will provide specific or parameterized task steps and control logic. These programs will use the robot's basic programmable functions, such as motion, video capture and motion detection, distance and height measurements, proximity sensing, robot posture and articulation, etc. These programming primitives may be organized into re-usable mission modules. Advanced programmers will have the ability to program new robot primitives and provide access to these as higher-level programming constructs.
- The Personal Rover will have an inherent obstacle and collision avoidance capability. This will include both emergency-stop behavior when faced with something dangerous (like a cerebellar reflex) as well as the ability to attempt to reach a goal position while moving in such a way as to avoid obstacles along the way and attempt to find paths around them. The obstacle and collision avoidance system will give the Rover robustness in its ability to use autonomy at appropriate times to manage the Rover's and the mission's safety. The Rover's programming capability will include the ability to parameterize and exploit obstacle avoidance and collision detection functions.

- A temperature probe for recording temperatures of objects the probe is in contact with (Readout in Celsius and Fahrenheit).
- Metrology Measurement Capability (Readout in Metric [centimeters/millimeters] and English) of height of object above robot's wheels. What about other distance measurements?
- A home base docking and recharging station, indicated by distinctive markers, including a self-docking capability.
- Ability to communicate remotely via wireless or infrared signal to upload/download data from programming/interface station.
- The Personal Rover knows its name and the names of others, so it can address various members of the household, friends, and pets.
- The Personal Rover has the ability to record various elements of its behavior, including capturing images, recording temperature and other sensor readings, and remembering various locations and proprioception and position data. These data are time stamped and sequenced, and may be referred to both during the mission and during post-mission data analysis. Recorded data can be supplemented with additional summary data like; temperature maximums and minimums, and distances traveled such as furthest. The Programming and Control Center application will support visualization and review of saved memories.
- The Personal Rover will be a unique toy in that it will possess a set of its own intents. These intents may be built in or updated via download from the programming station. These intents give the robot the ability to suggest activities and to pursue its own goals when given the explicit opportunity or as alternatives should planned missions become untenable. By expressing its own intents, the Personal Rover will invite children to try new things and continue to grow in their relationship to the toy.
- The ability to generate personalized reports of activities through analysis of its memories.
- The ability to instruct and challenge the user through execution of planned educational curricula.
- The ability to solicit the user for help in the midst of a mission.
- The ability to inform the user of mission status according to set of rules as to where and when to contact them.
- The ability to execute repeated tasks on a programmed schedule. This will include the ability to help the user doing the programming of potential

conflicts in the programs. Such scheduling will be extremely important in allowing the Personal Rover to have multiple programs from multiple users and the ability to sort out and prioritize the missions in the face of multiple, conflicting demands.

- The ability to learn knew actions to augment the set of built-in actions and behaviors. In effect this is a means of assigning prosody of actions for certain behaviors. This may require a lower-level programming interface. This allows the robot's behaviors to be personalized and take on the personality preferences of the user. This function could be supported in both a top-down and a bottom-up style of programming as follows. In top-down the user would specify that it wishes the robot to insert a new action behavior into its mix. The UI and robot would then take the user down to the next level and learn the required action. In the bottom-up approach the user would pre-define the sub-actions first, then combine those defined sub-actions at a higher level.
- An ability to reason forward in time about missions and requested actions. For example, power management predictions could help the user understand whether an assigned set of objectives is achievable with the available power. The Rover will be able to communicate the implications it derives from reasoning forward in time to the child in order to help visualize the implications of the mission planning and programming steps under development.
- The ability to emit meaningful sounds during execution of actions. These sounds could include pre-defined or recorded sound effects, as well as the means of synthesizing or generating sounds using wave form synthesis. In addition, the Personal Rover may include speech output - either as synthesized text-to-speech or through concatenation and playback of pre-recorded speech segments.
- The ability of the Personal Rover to produce other meaningful visual outputs, such as the flashing of LED indicator lights.
-

Principles of Operation

This section defines the Personal Rover's modes of operation, programming, and control from the user experience standpoint. These principles provide a basic vocabulary of operations, means of control, and things to do with the robot outside of the formal structure of a (tightly or loosely) structured mission. Beyond the basic product feature definitions of the previous section, these principles define various useful modes and skill sets that the product supports and helps users master.

MODES OF PLAY

The Personal Rover will support multiple modes of play. The multiple modes of play will provide engaging user experiences across a range of player needs and objectives. The basic modes of play include:

FREE PLAY

In Free Play mode the Personal Rover will engage in autonomous behavior under close supervision and possible interaction with the child. In Free Play mode the Personal Rover may engage in a combination of the following types of autonomous activity:

- Exploration - The Personal Rover will explore a small area of its environment in order to establish a safe area for movement and other activities.
- Searching - The Personal Rover may engage in specific searches for its recharging station or for its distinctive markers
- Showing Off - This will allow the robot to demonstrate its personality while showing off with movement tricks and dances.
- Motion Following - The Personal Rover can use its motion detecting video capability for camera and/or physical following of detected motion.

Free Play mode should also encourage the child to engage in some basic learning activities and begin introduction of transitioning the child towards instructional criteria, missions, task productivity, and agent programming.

INSTRUCTIONAL CURRICULA

The Personal Rover is an ideal platform for cooperative instructional curricula that teach science through interactions and activities with the toy. The Instructional Curricula play mode will serve two key functions. First, Instructional Curricula play mode will help the child learn about the functions and uses of the Personal Rover. This help system will evolve with the user's proficiencies, and will motivate the child to grow into the full depths of the product.

Instructional Curricula delivery and execution will also form the nucleus of an innovative teaching platform. The Personal Rover will be able to execute interactive lessons designed around specific learning objectives. These lessons may be tightly or loosely structured around supplemental teaching materials and objectives. The Personal Rover will use its evolutionary help system to work patiently with the child in fulfilling the objectives of the mission in the face of difficulties.

The Personal Rover product architecture and online community services will encourage and assist in creation of new instructional curricula lessons. These new lessons can be synched with or customized to specific teaching plans and objectives. For example, NASA could issue special missions, such as "Making a Temperature Map of a Distant Planet", that would teach robot owners how to use the robot's temperature probe and mission planning software to make a temperature gradient map of a small area. The product would encourage the child to put items of varying temperature within its experimental range. This activity could include follow-on lessons around graphing and analyzing the results of the temperature mapping experiment.

The following table summarizes several types of Instructional Curricula lessons. These lessons highlight the degree of structure imposed upon their successful execution, and show how both structured and unstructured approaches may accomplish the same teaching objectives.

Lesson	Description	Structure
Temperature Mapping (1)	Place hot and cold objects in a known configuration and have the robot sample temperatures in a square grid within an area. Plot a colorized map of the temperatures. Compare our temperature map to maps of temperature and other measurements made by the 2004 Mars Rovers.	Highly Structured
Temperature Mapping (2)	Have the child propose its own set of hot and cold items to map the temperature of. Suggest whether the robot can detect the residual heat from where a pet has lain.	Loosely Structured
Detecting Life (1)	The Personal Rover can suggest experiments to detect various types of biological processes in its environment. The lessons will integrate with the various techniques NASA will use to conduct biological experiments on Mars. The Personal Rover product may include auxiliary experimental equipment, such as pH paper or safe chemicals that will cause chemical reactions which the robot can help deposit, collect and measure.	Highly Structured
Detecting Life (2)	Have the child propose their own experiment to detect for the presence of life. Suggest that there are several ways to detect life; by detection of some chemical process, by detection of some residue of its presence, by detection of motion or response to stimulus, etc.	Loosely Structured
Power Consumption and Recharging	Help the child experiment with understanding how the Personal Rover consumes power. Suggest challenges that will help the child analyze and predict energy use, and develop strategies for	Both Highly and Loosely Structured

	recharging the robot via both solar collectors and the home base recharging station.	
Motion Detection	Help the child master use of the robot's vision system. Encourage experimentation with real time and time-lapse photographic techniques to capture the evolution of various phenomena such as plant growth, pet activity, or security surveillance.	Loosely Structured
Scientific Methods	Demonstrate the scientific methods NASA and other scientists use to assure that experimental results are accurately obtained and not contaminated. Teach the child about the delays in radio signals between controllers on Earth and Mars-based probes. Demonstrate the differences between use of the robot for data collection versus data analysis, and demonstrate how robots and people can work together to analyze data and draw experimental conclusions about hypotheses.	Highly Structured
Metrology	Help the child learn to use the Personal Rover's triangulation and distance and height measuring functions to build maps and models of objects in the robot's environment. Teach basic principles of precision, accuracy, and repeatability of scientific measurements, including use of and conversion to/from the metric system.	Both Highly and Loosely Structured

PRE-DEFINED MISSIONS

In addition to Instructional Criteria, the Personal Rover will be able to organizes its activities into Missions. Pre-defined Missions will allow the child to structure a set of robot movement, measurement, vision, and power management functions into a visual task schedule or flow chart. The Programming and Control Center software will help the child organize individual missions and to schedule the Personal Rover to pursue multiple missions during its day.

Pre-defined missions will be designed to help the child master robot programming and mission scheduling skills. Pre-defined missions will also relate to the scientific methods and proactive planning that NASA scientists use to accomplish objectives remotely via their Mars probes. Pre-defined missions will seed the child with ideas to pursue as they master the ability to program their robot to accomplish tasks for them.

USER DEFINED MISSIONS

Beyond Pre-defined Missions, the Personal Rover will be unique in its ability to encourage the child to formulate and execute their own missions. The robot's user interfaces will invite children to try new robot functions, and challenge them to learn about and apply new scientific methods toward the programming, experimentation, and analysis of their own missions. The user experience will be one of non-pressured encouragement. Children will be challenged to try new things periodically, but will not be forced into nor penalized for not responding to such suggestions from the Personal Rover or the Programming and Control Center software.

Both pre-defined and user missions will be directly supported by the Personal Rover web site and online community. Mission goals and mission programming elements will be exchangeable as Internet uploads and downloads.

SHARING OF MISSIONS AND CAPABILITIES

The appeal of the Personal Rover will likely get multiple family members and friends involved in its play. The mission scheduling software will allow multiple people to build and execute Personal Rover programs during the same day. The software will help all the team members understand how to share the robot's bandwidth and resources, such as power consumption, to permit the robot to pursue multiple objectives.

MULTI-ROBOT COMPETITION AND COOPERATION

Advanced features of the Personal Rover will allow two Personal Rovers to interact cooperatively or in friendly competition. Both Curricula and Missions can be structured around such cooperation and competition.

INTERFACE MODES

The Personal Rover will be unique in its ability to encourage its users to explore the many dimensions of its features and capabilities. This will be accomplished through interactive experiences that empower the Personal Rover to suggest and encourage its users to try new things. These user experiences will be suggestive rather than structured and forced - the user is not penalized for choosing not to act on a suggestion from the robot.

The following describe categories of interface modes that the Personal Rover will support in order to provide flexibility and likeability as an integral part of the user experience.

PRODUCT FEATURE DISCOVERY

The Personal Rover interfaces will be designed to provide encouragement to explore capabilities and functions of the product. This encouragement may come from either the robot itself or from an active voice as part of an interface metaphor. This could include requests for mission objectives from NASA mission control or scientists.

The user experience will be designed to adapt to the child's proficiencies and weaknesses. This will permit the Personal Rover to reveal features about itself at the child's pace for learning.

AUTONOMOUS BEHAVIORS

The Personal Rover will maintain a set of autonomous behaviors. These behaviors will serve to give the robot a sense of personality, and to give the robot alternative behaviors in the face of error conditions. The child may use the Personal Rover's Programming and Control Center to refine or augment the robot's repertoire of autonomous, prosodic behaviors with custom alternatives. A sound set of autonomous behaviors will permit the robot to exhibit a sense of smarts and self-motivated intent that can help encourage an active relationship between the robot and the child.

PATIENCE IN THE FACE OF ERROR CONDITIONS

Of particular importance is the Personal Rover's social interaction style to exhibit patience, understanding, compassion, and help in the face of challenging circumstances. The Personal Rover's interaction style as manifested through its various programming and communications interfaces (Programming and Control Center, Robot-contained lights, indicators, sounds, ands text-to-speech, and e-mail/messaging communications), will provide a positive and affirming experience designed to identify and rectify anomalous conditions. Such interfaces will be designed to identify and explain difficult circumstances, patiently allow the child to attempt alternatives to correct problems, and provide insightful suggestions in the face to dire situations to help reduce frustration and get the child re-aligned with a productive and pleasing course of experimentation.

Consider the following example of a Personal Rover's robust and encouraging dialog with its child programmer in the face of an impediment to its desired movement.

Child: Issues move command from the robot's current position to a new one down the hall. Halfway to its destination the robot encounters an unforeseen obstacle.

Robot: "May way forward is blocked. Do you want me to try and find a way around it, or do you want to steer me yourself?"

Child: Responds via the Interface, "Try and find a way around it yourself"

Robot: Moves left, searching with its vision and proximity sensors for the wall and the obstacle. "I can't go around to the left. Let me try the right". The robot continues its search around the obstacle to the right. "I'm still stuck, I think I might need your help. If you think you can steer me safely around this, give it a try. Otherwise we may have to backtrack and try this step of the mission programming again. What do you want to try next?".

Child: Chooses, "Let me steer you" from the interface software and takes over control of the robot under remote control. The child decides to push the obstacle out of the way.

Robot: "Do you want to override the collision detection program and allow me to push against this thing?"

Child: Chooses, "Override Collision Detection" form the interface. The robot moves slowly forward until it is up against the obstacle. The child tries to push on the object but the wheels won't budge.

Robot: "Either this thing is too heavy or I'm not pushing on the right place. Do you want me to try another location on the obstacle, try another route to my destination, or try something new?"

Child: Selects, "Try Another Location on Obstacle". The child steers the robot to the far right-hand side of the obstacle and again overrides collision detection to let the robot try a push.

Robot: "I still can't make this thing move. Do you want to try another location on the obstacle, another route to the destination, or do you want to try something new?"

Child: Chooses, "Try Something New". The robot notices the child has never attempted to crawl over a small obstacle.

Robot: "We can try to climb over this thing if it isn't too high. Let me check." The robot's camera and sensors go to work trying to analyze the height of the obstruction. After a few seconds the robot replies, "I think I can make it over this thing. It only looks like it's 1.9 centimeters tall. Follow the instructions on the screen to help me try and climb over this."

Child: The child follows a set of instructions on the screen that involve steering the robot and allowing it to elevate its front wheels sufficiently to clear the obstacle. The interface is like a wizard sequence that helps the child develop a program for surmounting the obstacle. Finally, the robot clears the obstacle and stops safely on the other side.

Robot: "OK, I made it past that thing I think. Now I'll proceed to the original way point you wanted me to get to before that obstacle diverted us."

The above hypothetical dialog demonstrates how the robot and the child can work together to overcome challenges and to explore new features of the Personal Rover.

STATUS REPORTING OPTIONS

The Personal Rover will be flexible and generous in its ability to report its status and to collect, file, analyze and communicate the data it collects. All Personal Rover interaction and measurement points will include a time stamp and location tag that lets the child know the time and place that each action performed was conducted. The Personal Rover will be able to record various types of measurements, including video capture, within the resources of its on-board memory. When within communications range with the computer running the Programming and Control Center software, the robot will be able to transmit measurement data for disk or memory storage on that computer. This will off-load data from the robot's memory to the Control Center, and permit the robot to collect more data on a mission than it could safely hold just on its own.

The Personal Rover and its Programming and Control Center software will support various forms of communication with the child. Such communications will often take the form of status messages from the robot to the child, and control messages from the child to the robot. Consider the following examples:

- The child programs a status report into the robot's current mission. According to the program's rules, the Personal Rover, in conjunction with the Programming and Control Center software, will compose an e-mail message to be sent to the child's account when the robot completes the mission. The e-mail will contain measurement data and photos taken during the mission.
- The child receives an e-mail from the robot reporting on the status of its mission. The robot reports that there is an unknown obstacle preventing it from reaching its first mission objective. In response to this anomaly the child sends its robot an instant message. The instance message tells the robot to give up on that mission for day.
- The child decides to synch up the robot to some of the activities on the Mars mission. Using the Programming and Control Center and by connecting to the Personal Rover web site, the child is able to connect the robot to telemetry control signals from NASA and Mars. Now when the Mars robot makes a

move, the child's robot will move in the same way. The interfaces let the child compare the status of their robot with the status telemetry data from Mars.

ROBOT LOCATIONS, MAPS, AND NAMES

The Personal Rover will use various mechanisms to locate itself within its environment, navigate between way-points, measure distances, and determine when and where it is safe to travel. This will involve cooperative exploration of the robot's environment between the robot and the child. The result of this exploration will be the formulation of maps that both the robot and child can use to facilitate future programming and mission objectives.

The Personal Rover will utilize distinctive visual markers (supplied with the product) as its principle means of triangulation and location. The Personal Rover's Programming and Control Center software will allow the child to assign meaningful names to specific way-points located by triangulation measurements between markers in fixed locations. Provided the markers have not been moved, and no significant or insurmountable obstacles intervene in the robot's path, the child may use these named places as programming primitives when building missions.

INTERACTION MODES

The following are a first cut at the types of interaction and user interface control elements required to program, control, and monitor the activities of the Personal Rover. These controls are designed to support the required experience and modes of play objectives. These will likely require dedicated User Interface facilities and will be developed fully in close cooperation with the UI Designer.

STATUS

Examples of things that will require a dedicated status readout:

- Personal Rover Power Levels
- Personal Rover Location
- Status of robot within an executing mission
- List of memories, telemetry, and recorded data
- Commentary (Consider the ability to use generative techniques to generate a running commentary of what the robot is doing)

TELE-OPERATIONS AND CONTROL

Here are some things needed to support tele-operations and remote control of the Personal Rover:

- What the robot can see right now (snapshot)
- Where the robot is right now and its status: the Rover's proprioceptive state - including such things as position of sensor boom, steering wheels, battery charge level, etc.
- Motion and steering controls
- Sensor stalk and effector control
- Camera control
- Interpolation control - How should the robot conduct its movements when asked to interpolate between pre-set way points?

MISSION PLANNING

Here are some elements of mission planning:

- The Personal Rover and the user can build up a library of missions to choose from. Some of these missions can be in the form of re-usable subroutines (get from charging station to the front door)
- Missions can be loosely structured to provide or suggest concrete objectives (independent of how those missions are to be accomplished - 'Find something warmer than body temperature')

- Mission planning will include some form of scheduling for when, and how often the mission should occur. This may include the need for schedule conflict resolution. Such conflicts could come from inconsistencies or overlaps in mission demands from the same user. Also, some scheduling conflicts may occur when the robot is shared amongst multiple users.
- Contingency planning - Are there secondary objectives and pre-programmed instructions for what the robot should do in the face of adversities that prevent execution of a mission or mission segment?

INTERACTION AND REPORTING

The Personal Rover will have a rich capability for interacting with the child through both real time and message-based reporting mechanisms.

Here are some aspects of interaction and reporting:

- The user needs to specify how the Personal Rover should report on its mission. This can include notions of information filtering. The robot records lots of things, but only select ones are presented to the user.
- The user needs to specify the communications channels and mechanisms and schedules for reporting. This implies that there are several reporting modes; real time, near real time (de-coupled by delays and communications overhead), and post-mission reporting.
- In addition the planned reports per mission, the Personal Rover can maintain state across missions. This data can be used to track history of certain activities (temperature measurements), and to establish records (maximums/minimums/firsts/lasts, etc.)
- Consider the power of a XML-based structuring of recorded data and robot memories. This would permit the following types of things. Records could be sorted and searched. Records could be linked between recorded mission memories in interesting ways. This data in XML format could be processed for publishing as HTML. This data could be used to seed a generative grammar engine that allows commentary and analysis to be made from the mission data.

- Reporting can include the ability to collect and format reports for publication. For example the mission data could be used to publish HTML. This publishing capability can include facilitating mechanisms for publishing accomplishments to a community web site.
-
- The Personal Rover's robust generative dialog mechanisms and XML-based report formatting and translating provisions will allow the Rover to synthesize reports to multiple audiences. The ability to use personality-based generative grammars will give the Rover a rich capability to describing its actions in ways appropriate to the audience. The Personal Rover may generate detailed reports in support of school science projects, personalized instant messages of current activities, or entertaining transcripts of mission activities to be shared with family and friends.

ERROR HANDLING

The Personal Rover will require robust mechanisms for error handling and reporting.

- The Personal Rover will be robust, persistent, and assiduous in its efforts to self-correct in the face of errors.
- Direct reports of problems from the robot through sounds and actions.
- The ability for the robot to 'get safe' by posturing and moving to positions that minimize interface and damage when stuck mid-route.
- The robot's error reporting will attempt to highlight and delineate error conditions and perhaps potential solutions (that require user intervention) or ways to avoid such errors in the future.

USER INTERFACE METAPHORS

The user experience should address the notion of who the user is interacting with through the UI. Such user interface metaphors will explore the use of additional voices and characterizations that the child interacts with through the Programming and Control Center application in addition to direct communications with the robot's personality.

The following represents a preliminary set of suggested metaphors for such interactions. Exploration of these metaphors will necessarily require close cooperation with the UI designer.

A successful user interface metaphor and third-party voices or characterizations must support the following:

- Suggestions to the User
- Basic Help Facilities
- Mission Planning - Accounting for Robustness and Error Handling
- Educational/Curricula Support - Concept Mastery - Product Learning Curve
- Scheduling
- Reporting and Messaging

There are advantages and disadvantages to introducing another character or voice besides the Personal Rover's to the interface paradigm.

- Advantage - Educational demands not coming from the robot if child does not take to education
- Advantage - Suggestions and corrections and user errors are not scolded by the robot
- Advantage - Interface agent can believable know and do things that the robot might not be able to do believably.
- Disadvantage - The robot may not seem as smart when the third party character is required to interject new ideas and solutions to problems.

The mission examples given later in the document utilize a 3rd character in the guise of a NASA scientist. This character is designed to make explicit the relationship between the NASA Mars mission and the use of the Personal Rover as a learning platform for life sciences, planetary sciences, mission planning, and data collection and analysis missions and curricula.

PROGRAMMING IMPLICATIONS

This section describes the general facilities and mechanisms available for programming the Personal Rover. These programming mechanisms include both the use of existing programming primitives to build up useful programs, as well as the ability to extend the repertoire of programming primitives through the specification or synthesis of new programming elements. The overall user experience will emphasize a "No threshold, no limits" approach as advocated by Seymour Papert, which is also the foundation for the popular and successful Logo programming language designed as an introduction to programming accessible to children. "No threshold" implies that children will be able to begin programming tasks immediately through direct manipulation. "Not limits" implies that by recombining and building upon previous programming successes, the child can extend the robot's capabilities in novel and unforeseen ways.

TEACHING PROGRAMMING BY DOING

The Personal Rover will be interesting in its ability to teach basic programming tasks by example. Children will be introduced to programming immediately through the Personal Rover's ability to learn its program by user training. In user training the child moves the robot and indicates how to use its camera and sensors by operating the root tele-remotely via the Programming and Control Center software. The robot remembers the key steps in this control process, and uses them to build a program of actions. The steps in the robot's programming show up as programming module steps on the Programming and Control Center interface. The child is learning programming through direct manipulation of the robot, and need not memorize or master a set of traditional programming constructs or languages to get started.

BUILDING ON PAST PROGRAMMING TASKS AND PROFICIENCIES

As the child masters various programming tasks, they are saved as re-usable programming modules. Many programming module steps include user tunable parameters that customize basic robotic functions to specific situations. After mastery of programming by example as described above, the child can being to learn and experiment with the synthesis of new programs from a library of basic robot capabilities. Each new mission and mission stage that the child builds through their programming efforts augments the available programming library. Use of elements from the programming library also encourages the user to learn new types of programming logic. Forms of programming logic include if..then..else and repetitive looping constructs that allow the user to specify what the robot is to do in the face of certain conditions, and to perform tasks repeatedly for a fixed number of iterations.

EXTENDING THE ROBOT'S REPERTOIRE: PROSODIC ACTIONS

The Personal Rover's programmability extends to all facets of the robot's operations. An example of the role and utility of augmenting the Personal Rover's repertoire is the specification of expressive or prosodic actions. When the Personal Rover performs an action we may consider that action to consist of two parts, a goal or intent (the 'what'), and a means of expressing the action (the 'how'). The behaviors the Personal Rover exhibits can be considered as a sequencing of logical intents in concert with appropriately expressive prosody that reflects to the robot's observers the task and/or mood behind the actions.

The robot's interaction and perceived user experience can be enhanced by providing expressive mechanisms that give personality to and suggest anthropomorphic meaning for its actions. Consider the basic capability of robot locomotion from point A to point B. The root action here is captured by the basic intent; "Move from A to B". This specification does not elaborate on the posture of robot during the move, nor does it specify the style of locomotion to employ. Let's assume a default posture for the robot during movement, and assume a direct path and nominal speed for the locomotion. Now let's look at how this basic motion action can be customized for expressivity by changing the robot's posture and motion characteristics.

We can take this same basic 'bee-line' movement and add prosody as follows. First, we can modulate the height of the sensor stalk. We can slow the rate of motion up or down from the nominal speed. We can also take liberties in having the robot's course deviate from the straight-line path. The following table summarizes a series of anthropomorphic actions obtainable from various combinations of sensor stalk and motion. These could be further refined through the use of pauses in the motion and suggestive sound effects.

Intent = "Move from A to B"		
Prosodic Intent	**Sensor Stalk Position**	**Movement Characteristics**
Officious	Medium	Medium, Straight
Curious or Exploratory	Medium to High	Slow, Zigzag
Sneaky	Low	Slow, Circuitous
Busy	Medium	Fast, Straight
Happy or Fun	Variable	Variable

The above example shows how combinations of the robot's basic posture, movement, (and event sound primitives) can be used to synthesize a rich and anthropomorphically meaningful prosody language. Employment of these mechanisms provides an additional channel of communications between the robot and those observing it, and provides a rich platform for enhancing the overall user experience. The off-the-shelf Personal Rover will have a built-in repertoire of expressive actions. The robot will also encourage users to augment this basic action library with their own personally tailored prosodic actions. Finally, the ability to string such actions together in series suggests a rich palette for synthesizing rich behaviors that help the user and robot communicate about what is happening through interesting and expressive combinations of how a task is being done.

PROGRAMMABLE AT EVERY LEVEL

The Personal Rover will be programmable an extensible in an open source fashion at every level of its implementation. Consider the following tentative product programming architecture. The architecture forms a logical hierarchy of functional complexity that maps onto appropriate instructional applications programming environments.

- Drivers - The Personal Rover will user a standardized device driver architecture to permit software modifications and extensions to the set of fundamental device drivers used in the robot's systems. Note that the Personal Rover product architecture could contain additional memory or I/O ports (RS-232, USB, etc.) that experimenters can use to add robot extension devices to the basic chassis.
- Operating System Functions - The open source initiative may elect to open the Personal Rover's operating system kernel to extensions. These extensions would effect the real time process scheduling, device control, and I/O functions of the robot.
- Robotic System Functions - Likewise, the open source initiative will permit advanced experimenters to manipulate the parameters of the Personal Rover's built-in sensors and controllers.
- Vision System Functions - The open source initiative will encourage developers to augment the basic visions capabilities of the off-the-shelf Personal Rover with additional image processing functions.
- Robot Actions and Interactions Functions - The postures and animatronic/prosodic movements of the Personal Rover can be augmented and trained by the user. This permits the user and the robot to develop a personality and customize style of operation.
- Primitive System Functions - This interface will allow users to build new custom robotic system function primitives. These primitives become integrated into the Personal Rover's 'vocabulary' of know primitives.

Custom primitives can subsume other custom primitives, making extensions to the vocabulary rich and powerful.

- Reusable Programming Module Functions - Users can build their own higher level programs and package these into reusable modules. These programs allow users to build libraries of useful utilities. Similar to Java Beans, these modules will have parameters that allow them to be exchanged and instantiated from one user to another and incorporated into the new user's physical and programming facilities.
- Higher-Level Robot Programming Languages - Finally, the Personal Rover's programming architecture will encourage and facilitate the ability for developers to build their own higher-level graphical and non-graphical programming language front-ends to the robot's internal programming language. This architecture will facilitate the building of translators from programming language visual and written syntaxes into the robot's machine language primitives.

Personal Rover Missions and Scenarios

This section outlines a script of prototypical missions and scenarios that can be realized from the toy's basic capabilities. These mission scenarios are not exhaustive, but serve to illustrate a set of basic 'use cases' that give a representative overview of the Personal Rover's usefulness as a successful toy product that exhibits strong synergy with the goals of robot science and exploration of appeal to the product's sponsors. This section will present a series of believable stories that relate the capabilities of the Personal Rover to the experience of those programming and using the toy.

Jenna Gets a Robot

I'm Jenna and I'm 11. Today's my birthday! I just opened the best present, my own Personal Rover. Dad helped me get the robot all set up and recharged. Now we're ready to play.

I wonder how smart this robot is? I turn it on and it jumps to life. The video head raises up and looks around. Then it talked.

"Hi, I'm your new Personal Rover. Watch this!"

The robot left the recharging station and began to snoop around, going out and back in little bursts, like it was seeing if it was safe.

"I can dance", the robot declared. Then it proceeded to move very fast, forwards and backwards, even standing up on two wheels.

"Hold out one of my special reflectors in front of me so I can see you."

I took the shiny yellow marker and held it in front of the camera. The camera moved a little, then the robot stood straight up and said, "There you are! Let me take your temperature. Hold your finger against the bottom of the temperature probe where the green light is blinking". I stuck my finger under the plastic probe and waited.

"Yep, I detect you - Did you know that you're 28 degrees warmer than the air in this room? Now move the marker in slowly in front of me and I'll chase it". As I moved the marker first the camera, then the whole robot began to follow my every move. If I went to fast the robot would excitedly look around until it again found me, like a cat at play. This thing is smart.

JENNA LEARNS TO PROGRAM

While dad and I played with the robot, mom installed the Personal Rover's Programming and Control Center software on my PC. "Here Jenna, we can tell the new robot what we want it to do". I took the mouse and pretty soon figured out how to view the robot's vital signs. The screen showed a skeleton of the robot, with motors, wheels, wires, sensors, and the camera. I could move parts of the robot, like the neck, on the screen, and the new robot would move the same way.

Then I switched to control mode. I could drive the robot forward or backward, and make it turn. I could adjust the camera and see the new picture on my screen. I could steer the head with the camera in one direction, while sending the robot's body off in another. I thought the robot would get dizzy.

"Jenna, if you go to program mode you can make the robot remember how to do things you command it to", mom said.

I changed to Program Mode. There was a scientist in a lab coat on the screen. "I'm Doctor Lamberts of NASA's Mars Rover Department. I see we have a new robotics researcher on board. Congratulations, and

welcome to the team. I know you'll make important contributions to our Mars mission with your robot here on earth.
"Now, let me introduce you to your new Personal Rover. I leave you two alone while you get to know each other".

The Doctor left the screen and I was looking at computer generated version of the robot on the screen.

"Hello", said the on-screen robot. At the exact same time the real Rover also said, "Hello".

"If you tell me your name we can be friends." I typed in "Jenna".

"Hello, Jenna. You're my new programmer. I'll teach you about robotics and programming if you teach me about the real world. "I'm programmed to respond to you by name. Only I don't have a name yet. Why don't you pick one for me?". This was wild, I had to pick a name. I thought for a while, then it spoke again."

"What's the matter, cat got your tongue? Can't you think up a cool name for a crazy little robot like me? You can always change it if you don't like it, and I'll have no hard feelings". I hesitated twice, but then typed in 'Deek' because it was some kooky astronaut's name I heard about in Miss Lincoln's class.

"OK Jenna, call me Deek. Deek. Deek. I like that, thanks. Now Jenna, I'd like you to help me find my way around a little. Put me in the recharging station - that's my home base. When we first learn to program, all our missions will start from this home base." I plugged the robot into the base.

"Now place a marker on its stand in front of me, Jenna". I went across the room and put a shiny marker on its stand facing the robot. The robot camera moved, and the robot came slowly toward the marker. When it got a few inches away, it stopped.

"OK, found it! Now put another marker on its stand to my right, and another one to my left." I put two more markers out like the robot suggested. Then the robot camera moved a lot, first to one marker, then the next, then the next. Then the robot moved in short bursts forward, back and turning, until it was in the middle of all the markers.

"Jenna, I've used triangulation to locate myself in the exact center of the triangle of markers. I'm 26 inches from the first marker, 26 inches from the marker to the right of that one, and 26 inches from the other marker on the left. Now use the control instruments to move me somewhere else within the marker triangle". I used the mouse to drive the robot back toward the first marker. I used the buttons on the screen to tell the program I was done moving.

"I've taken visual distance measurements from the markers so I can find this spot again. Now I'm 4 inches from the first marker, 40 inches from the second marker, and 42 inches from the third marker."

Now Doctor Lamberts came back onto the screen. "Jenna, Deek has remembered the location of his new point. On the screen you can see some boxes connected with lines. The boxes are programming steps, and the lines indicate the sequence of execution of those programming steps." I saw what he meant. The first box said, "base station", the second box said, "Midpoint", the third box said "experiment point 1". Between the boxes were lines with arrows that pointed from the base station to the midpoint, and from the midpoint to the new point I had steered the robot to.

"I'll return to the base station and we can test your new program", said Deek. Deek re-docked himself and after a few seconds of being still left the docking station and crawled directly to the midpoint. It used its camera and moved back a forth a couple of times until it was perfectly centered. Then it moved toward the experimental point that I had moved it to. Again the camera and wheels did some searching to position the robot exactly. At each step along the way the box on the programming screen turned green to let me know what the robot was doing.

The doctor appeared next to "Congratulations Jenna, you've made and executed your first robot program. Now Deek knows how to get to experiment point 1, and you can use this new program as a self-contained module to get from the base station to this point anytime in the future. Provided of course that the markers are in the same places. Let's try another experiment that will teach you even more programming skills…"

Deek Teaches Jenna Science

I played with Deek every day after school. I learned all about programming Deek to go places and do things. I had to get mom to agree on where I could place Deek's markers so he could find his way around. I identified three special areas for Deek and I to experiment in. I named them after three of the experimental sites on NASA's Mars Mission - Alpha, Beta, and Gamma. I learned those words were the first three letters of the Greek alphabet.

Deek and I went through the mission book and software, and did some neat experiments. I made a temperature map of the kitchen floor, and put some ice cubes and cup of hot water on the floor to see how they effected temperature. The robot took measurements every 5 minutes, which I programmed into a mission to make a new temperature map and compare it to the previous one.

After 10 measurements the Control Center software helped me make a color map of the kitchen floor's temperature, with the cooler parts in blue and the hotter parts in red. The ice made a blue eye-shape on one side of the map, and the mug of hot water made a round red eye-shape on the other. I could make cool graphs of the temperatures as rings of the same temperature that I learned were called "isotherms". I could even watch a short movie of how these maps changed over time, the blue of the ice slowly turning green like the rest of the room while the red of the hot water mug cooled also toward green. I was able to compare my temperature map with those of other kids and to even compare it to temperature maps being made right now by the robots on the Mars Mission. I turned this mission in as a report for science class and got an A!

Jenna Contributes an Advanced Mission to the Web Site

I also logged on to the Personal Rover web site. I downloaded a cool "skin" for the instruments on my interface. They have the same look as the controls that NASA uses to command the Mars Rovers from Mission Control. I also downloaded some pictures and missions that other kids had put up on the site.

I want to make a cool mission that I can put up on the site too. I'm getting pretty good at programming the Rover and building missions. Yesterday Deek asked if I wanted to try programming a mission to do a job for me. I said no then, but now I want to build a mission to have the robot check the level of dog food in Missy's dish. I'm 'sposed to do that but I forget if I have to look more than once a day. So I want to make Deek go out once every hour and check the level.

I always check the robot's status page on the Programming and Control Center software when I begin programming a mission. I feel like a doctor checking up on her patient before an operation. Deek is checking out perfectly now that I cleaned one a proximity sensor that was reporting an error. I've already placed markers around Missy's bowl so Deek can find it. Now it's time to use the programming and mission control screens to build the mission.

First, program Deek to go to Beta Area. It's easy. I just drag and drop the Beta Area programming module that Doctor Lamberts helped me build. I tell Deek to execute the first part of the program, and he obediently drives himself to the Beta Area. Next I flip to control mode, so I can drive Deek from the Beta Area to Missy's food dish. I have to do some tricky driving over the small step down into the laundry room. But Deek is smart enough to go up and down small steps. Sometimes he looks funny, sticking his head way forward or way back to help shift his weight and get the wheels to go up or down.

My program is growing. On the programming screen I've already added several more programming steps to get Deek exactly positioned in from of the food dish. Now I have to find a way to detect the level of Missy's food. This is like the problems the robots on Mars have to solve - how to detect the presence or absence of a particular thing. Neither the temperature or proximity sensors will help here. There must be a way to detect the food. I think for a while. I've got it! The markers. I'll stick a marker in the bottom of the bowl. Then I'll have the camera hover over the bowl and try and see the marker. If I can't see it, or see it just barely, then I'll assume there's still food covering it. If I can see the marker strongly, I'll know Missy's eaten all her food.

I put the marker in the bowl and cover it with food. I had to lock Missy outside because she wanted to eat when she heard me pour the food. I added several programming steps to my mission, driving Deek right up to the bowl, then moving the camera head to point straight down into it. I added the marker detector programming module. I had to connect two other modules to the detector module. First, what to do if I saw the marker. The second, what to do if I did not see the marker.

The cool thing is that each time that Deek takes a measurement, he also records the time the measurement was taken. It takes me a while, but I add some programming modules that let Deek report back to me about the food level in the bowl. Depending on the time of the detection I tell Deek to inform me either by e-mail, or by telling me. If the food dish is found empty while I'm at school, I have Deek send me an e-mail report. Otherwise, I have Deek remember to remind me about the food dish the next time I turn him on.

Now I have to plan the timing for the mission. I use the mission planning scheduler to tell Deek what times to check for food. I have him check every hour from 9 in the morning to 4 in the afternoon. He'll return to the charging station after each exploration, so he won't run out of power. Using the mission scheduler I can even have Deek execute different missions at different times.

Next I tested the mission program. First I made sure the marker was covered and let Deek find his way from the recharging stations, to Beta Area, and then into the laundry room. It handled the step with ease. When it go to the bowl the camera hovered over it.

"The dog does not need feeding", appeared on the screen. It worked! Deek knew that the dish was full. Next I emptied the dish and returned Deek to the recharging station. Again Deek found Beta Area and looked into the dish.

"Missy needs food", appeared on the screen. Wow. Now I can have Deek look after Missy's food dish. I couldn't wait to upload my mission to the Personal Rover web site. I followed Doctor Lamberts' instructions and saw my mission on the Internet. I instant messaged my friend Julie who has a Personal Rover named Cara, so she could download my experiment and try it too.

DEEK ANSWERS THE DOOR, AND MAKES A MOVIE

My Aunt Bettie came to visit. She taught science for 12 years, so I wanted to show Deek to her. I made a special mission so that Deek would greet her at the front door. I used a motion detection programming module so Deek could tell when the door opened. Of course my dumb brother Eddie kept opening the door to trigger Deek. It was fun. I used the Robot Actions programming menu to program Deek to do some tricks. I made him nod very fast and waggle his neck. Then I had it follow a marker that I was going to give to Aunt Bettie when she came through the door. Deek came to life right as Aunt Bettie came through the door. Deek did his greeting dance and then followed her into the kitchen to meet mom.

Aunt Bettie loved Deek, and said she was so proud of me for learning how to make a robot do so many things and learn so much about Mars and science. During her stay she helped me set up some bean sprouts growing on a wet paper towel. We had Deek take a picture of the seeds every half- hour. I could tell the seeds were sprouting, but it was so cool to see the time-lapse movie that Deek made. I could see the roots twist and turn, and the stalks stretch upward. I was able to measure the seed's growth rate, and made that into another extra-credit science report.

JENNA AND DEEK CONTINUE TO GROW TOGETHER

Doctor Lamberts has helped me keep a diary of my experiments with Deek. The diary shows the robotic and programming skills I've learned, as well as the science experiments and missions I've made and executed. I've printed some of them out below to show people.

Mission	Description	Sample Measurements
Guard my Bike	Have the robot position itself to watch a given location. Use motion detection to determine whether there has been a change in the location being guarded. Report any changes at the location to Jenna via e-mail.	Mission Execution Begins - Mission = "Guard my Bike": 10:00 AM Deek on Station: 10:06 AM - On Guard Motion Detected: 11:13 AM - Sending picture ia e-mail for Jenna's review
Look for Life	Have the robot measure the height of a bamboo plant. Take measurements 4 times a day. Take a snapshot of the target object with each height measurement. Report any significant changes	Mission Execution Begins - Mission = "Look for Life": 9:00 AM Deek on Station: 9:12 AM - Taking Measurement and photo. Height = "9.2 Centimeters" . . . Deek on Station: 12:32 PM - Recharging using Solar Panels. Deek on Station: 2:00 PM - Taking Measurement and photo. Height = "9.3 Centimeters". We may have detected evidence of life!"
Greet Jenna After School	Have Deek station himself by the front door at 4:30 PM. When it detects Jenna holding a marker after detecting the door open with motion detection do a greeting dance.	Mission Execution Begins - Mission = "Greet Jenna After School": 4:20 PM Deek on Station: 4:23 PM - No Motion Detected.

		Deek on Station: 4:37 PM - Motion Detected - Looking for Marker. Deek Ready to Greet: 4:37 PM - No marker detected, probably just your dumb brother. Deek on Station: 4:39 PM - Motion Detected - Looking for Marker. Deek Ready to Greet: 4:39 PM - Marker Detected - Hello Jenna, long time no see. I have so much to tell you about what I did today. Greeting Dance initiated.
Mars Experiments	Have Deek take temperature and sensor readings that mimic what the Mars probe is sensing today. Compare the Mars data with the mission data Deek Collects	Mission Execution Begins - Mission = "Mars Experiments": 7:00 PM Deek on Station: 7:01 PM - Beginning Data Collection. Deek on Station: 7:01 PM - Taking temperature reading = 22° C. Mars reports temperature reading = -77° C. Deek Moving to Next Measurement Point: 7:17 PM. Deek on Station: 7:18 PM - Taking temperature reading = 22° C. Mars reports temperature reading = -76° C.

I've got to go now. Tomorrow Mom is taking me to the Community Center where there's a Personal Rover show and competition. I'm working on a program to have Deek mimic a strange life form on a far away planet like Mars. I downloaded a cool program from the Personal Rover web site that lets me program animal behaviors and needs into the robot. When running that program Deek needs to find energy in the form of solar power, shelter in a safe corner of a room, and has to avoid cold areas where I've put ice out on a plate. I changed the program so that Deek needs to find a warm spot under a desk lamp that I put on the ground. Deek uses his temperature probe to make a temperature map to find the warm spot. I can even move the lamp around and Deek will still find the spot. Sometimes he has to go way around the ice plate. Deek almost seems alive the way he avoids cold and goes towards heat. I'm going to show moff this mission tomorrow at the Community Center robot show.

User Interface Requirements Summary

This section consolidates the user interface implications suggested by the descriptions in the document. This set of interface implications and functional use cases serves as a checklist for conceptualization of supporting user interface elements by the UI Designer.

STATUS SCREEN

The Status Screen provides a synoptic view of the Personal Rover's systems and sensors. It provides access to parameters used to adjust and tune the robot's functions. It also provides a limited direct-manipulation interface that allows the child to verify proper operation of the robot's basic functions and control servos and motors, and camera.

TELE-OPERATIONS SCREEN

The Tele-Operations Screen provides controls to manipulate the robot through motion and functional activation of its various sensors and devices, including the temperature probe and camera. The screen provides controls for locomotive movement, acquisition of visual markers, prosodic and metrological positioning of the robot's head and sensor stalk, and advanced motion capabilities that can help the user conquer rough terrain and obstacles.

MISSION PROGRAMMING SCREEN

The Mission Programming Screen helps the child build programs that specify the stepwise actions the robot is to take in order to accomplish a given mission. This screen provides access to a library of pre-defined and user developed program modules that may be integrated into mission programs under development. These programming modules provide for basic robot movement, sensor, and camera controls. Examples of such programming modules include:

- Movement commands - Move to specific (X,Y) location relative to current position or in relation to triangulation between designated visual markers.
- Marker Acquisition - Search commands that direct the camera head to find and acquire the location of all visual markers with the robot's articulated field of vision.
- Picture Acquisition - The ability to take photo snapshots of a given aim point.
- Motion Detection - The ability for the robot to detect real time changes in its image that represent motion within its current field of view.
- Follow Commands - The ability of the robot to combine vision and motion commands to keep the camera tracking a moving visual marker, and the ability for the robot itself to move so as to follow a moving marker, if possible.
- Temperature Sensor Commands - The ability for the robot to take and record a temperature reading at its current location.
- Proximity Sensor Commands - The ability for the robot to record the absence or presence of obstacles in the field of view of its proximity sensors at the robot's current position.
- Action Commands - The ability for the robot to execute a pre-programmed or stylized set of movements. These movements may include prosodic actions that use the robot's posture, movements, sensors, sounds, and text-to-speech to add personality and informed user feedback to its actions.
- Logic Commands - The ability to give the robot decision making power about what to do in the face of perceived mission conditions. Logic commands give the robot elements of intelligence to make decisions, to repeat patterns of movements, camera angles, and measurements, and to recover and retry actions in the face of detectable error conditions.

MISSION SCHEDULING SCREEN

The Mission Scheduling Screen allows the user to manage multiple mission objectives during the robot's day. This includes the times to repetitively execute a fixed mission, and to highlight times when the robot has been asked to do more than one thing at the same time. Mission Scheduling helps the child understand the importance of planning and management of resources. Scheduling can also be used to establish times and means of status reporting so that the robot and child can stay in touch with one another as missions progress.

DATA COLLECTION AND ANALYSIS SCREEN

The Data Collection and Analysis Screen provides a means of visualizing the sets of data that the Personal Rover collects during its various missions. This screen may include capabilities to sort, search, and graph data points, maps, and time lines.

PERSONAL ROVER STATUS REPORTING SCREENS

The Personal Rover Status Reporting Screens provide a means of structuring the types of status data the robot is to collect, and to specify the status reporting communications channels the robot should use to record and communicate results. This screen can include configuration of Internet and Wireless messaging communications options such as emails, instant messages, and wireless SMS messages for communicating changes in or results from ongoing missions.

PERSONAL ROVER WEB SITE INTERACTIONS

The Personal Rover Web Site Interactions screens would show prototypes of the basic community features available. Examples of such community features include:

- The ability to upload and download mission programs so they can be shared with others.
- The ability to upload and download instructional curricula developed specifically to teach elements of robotics, mission planning and programming, and life and physical sciences.
- The ability to chat and read or post messages to the robot community about various topics of interest.
- The ability to view ongoing virtual and real world community activities, such as Personal Rover gatherings and shows in the user's area.
- Real time updates that correlate use of the Personal Rover with NASA's ongoing Mars Probe missions.

Next Steps

This section summarizes this report and offers suggestion of how this information can help in refinement of the product definition, user experience, and user interface prototyping efforts going forward.

This report has attempted to outline elements of the Personal Rover user experience. This report has tried to relate the basic capabilities of the Personal Rover product platform to educational and productivity opportunities gained from learning to program a robotic agent. The user experience has been designed to support both structured and open opportunities for children to learn sound scientific methods and principles in a fun and challenging way. The user experience has focused on personality-based interactions that affirm the child at every turn, employing smart help systems and suggestions to motivate the child to pursue new avenues of learning, and demonstrates how personality-based interfaces can use infinite patience and flexibility in helping the child reap the benefits of this learning toy at their own pace and on their terms.

The next steps for this report are as follows:

- Validate the product architecture and Personal Rover features suggested in this document.
- Validate the user experience approach, especially the voicing and characterizations used in the personality-based interface components of the experience. This includes vetting of the concept of a 3rd-party interface voicing and characterization as a user interface metaphor that serves to de-couple the Personal Rover from the instructional and error handling requests that may arise from the product's use.
- Provide the User Interface Designer with a conceptual framework and story from which to synthesize a set of prototypical interaction interface and screen designs.
- Finally, to motivate all members of the Personal Rover team to further visualize the potential and possibilities of the robot as a learning tool.