Resource-Efficient Methods for Feasibility Studies of Scenarios for Long-Term HRI Studies

Nate Derbinsky
Computer Science and Engineering Division
University of Michigan
Ann Arbor, MI, USA
nlderbin@umich.edu

Wan Ching Ho, Ismael Duque, Joe Saunders,
Kerstin Dautenhahn
Adaptive Systems Research Group,
School of Computer Science
University of Hertfordshire
Hatfield, United Kingdom
Email: {*w.c.ho,i.duque-garcia2,j.1.saunders,*
k.dautenhahn}*@herts.ac.uk*

***Abstract*—Long term HRI studies can be costly, firstly in terms of researcher time, hardware/software development time, data-collection, data analysis, trial preparation, trial execution, robot time and subsequently, in terms of funding for robotics and other equipment. Methods which reduce such costs by using resource-efficient feasibility studies to analyze study methods and propose outcomes, debug code associated with data collection/analysis, and sanity check human-robot interactions by simulating, predicting and generating feasible scenarios would therefore be welcome. This paper proposes such methods and provides physical implementation details of these methods in practice and data from a preliminary study.**

***Keywords*-feasibility studies; experimental methods**

I. Introduction

Whilst many large research projects propose that humans interacting with robots is achievable given existing robotic technologies and research efforts dedicated to human-robot interaction (HRI), only a relatively small amount of long-term studies have been presented/published in this field. Here long-term studies refer to a series of sustainable experiments that involve one or more human users repeatedly acting/working together with robots over an extended period of time in a complex environment and with a large repertoire of human-robot interaction. Long-term studies are indeed very desirable in investigating various aspects of HRI, such as the key features of a service robot, users' perception and reaction to robots, and scenario and methodological design of HRI, among others. So, in general, what makes the long-term use of service robots in such studies difficult?

One of the major problems for long-term studies of HRI is the resource cost associated with developing such studies. Costs may include researcher time (e.g. developing study materials), hardware and software development time (e.g. building/modifying the robot, coding, data-collection and analysis routines), data-collection time (e.g. preparing for and executing trials), robot time (e.g. time sharing between studies), and funding (e.g. recruiting subjects and purchasing expensive equipment required for studies with complex service robots). To give an example, typically the development of a service robot's functionality (such as the Care-o-Bot 3) in complex interaction scenarios and involving a variety of tasks requires a team of researchers to work on it for several years. If hardware needs to be developed/modified the time frame can be significantly longer until actual real-time studies with users are feasible. If the results of these eventual studies point out flaws in the scenarios that had been implemented, then typically modifications cannot be made easily without again involving significant costs. Such methodological issues have been acknowledged to significantly limit the advancements of the field of HRI ([1], [2], [3]).

Some of these costs may be reduced if resource-efficient feasibility studies can be run to analyze study methods and proposed outcomes, debug code associated with data collection/analysis, and sanity check subject-robot interactions. In our research we are particularly interested in long-term HRI studies, in which we investigate human-robot interactions over weeks and months within a smart-home environment, and it is in these cases that the pay off of feasibility studies could be dramatic.

In this paper, we discuss and evaluate methods that we have been developing to perform resource-efficient feasibility studies in this context. We begin by discussing requirements for these methods. We then introduce methods we have been developing at the University of Hertfordshire Robot House [4], a domestic residential home environment allowing for user-centered HRI scenarios in a familiar and natural context. For each method, we describe the associated goal in terms of reducing the study's resource utilization, analyze the degree to which it meets our methodological requirements, present preliminary evaluation data, and discuss future directions. While these are preliminary methods and findings, we hope to spark a discussion within the HRI community dealing with effective methods for feasibility studies that can improve the efficiency of such methods in our research.

II. Requirements for effective methods

For methods to be effective in feasibility studies for long-term HRI studies, we argue that they must jointly satisfy the requirements of (R1) *resource efficiency* and (R2) *outcome-relative fidelity*.

R1 - Resource Efficiency

The central goal of a feasibility study is to identify relevant issues without expending a great deal of resources. Thus, the method itself must not induce a resource burden. Methods should cheaply, quickly, and broadly produce outcomes that are relevant to the target study.

R2 - Outcome-Relative Fidelity

If a feasibility study is to be useful, its outcomes must be sufficiently accurate and trust-worthy as to support potentially costly design decisions related to further feasibility investigations, as well as the target study. The degree of accuracy necessary will depend upon the types of issues to investigate, but the method should produce outcomes that are informative.

 ISBN: 978-1-61208-250-9

Previous research in HRI had already recognized the need to perform feasibility studies before the completion of the final system. Several methods have been adapted from Human-Computer Interaction to HRI studies, including mock-ups and props [5], video-based [6] and theatre-based [7] methods, and Theatrical Robot [8], or Wizard-of-Oz studies [9] which can be applied once a prototype of the system is available that can be remotely controlled. However, mock-ups for complex robotic systems and their interactions with people are not easy to design, while theatre- and video-based methods lack the situatedness of the user in the interaction context (users are removed from the context by just watching either a video or live-performance). This article presents a method that replicates the situatedness and context of the user embodied in the scenario, despite the lack of a physical embodiment of the robot and its functionalities. Situatedness is an important aspect of embodied cognition and interaction with the physical and social world (e.g. [10]).

III. Methods

In this section we discuss and evaluate three methods we have used for feasibility studies at the UH Robot House.

A. Sensor Logging

Laboratories that design and pursue multiple ongoing HRI studies can quickly find themselves awash in the details of sensors. In addition to installing, testing, and writing software to operationalize sensors, a great deal of time can be spent on trying to unify diverse data formats, unsynchronized timestamps, and decentralized/disorganized log files. To address these concerns, we adopted three main components to our sensor-logging method. First, we centrally log all data to a relational database. The benefits to this approach are numerous: reliability, efficiency, and scalability when dealing with large numbers of sensors, frequent updates, and long-term studies; general data access and query methods, standardized via existing APIs (asynchronous, local or via network) and the SQL query language; synchronization of log values and timing via atomic, consistent, isolated, and durable (ACID) transactions; and flexibility to incorporate arbitrary sensor types/values and associated meta-data.

The second component of our sensor-logging method is sensor registration, which means that we explicitly associate meta-data about sensors (e.g. name, type, location, etc.) with each logged value. This registration is accomplished efficiently via tables and many-to-one relations in the database and can thus be queries later via SQL. This registration process directly facilitates mixed real-virtual sensing; we discuss and exemplify this concept in the Robot Modeling section, but in short, integration of virtual sensors allows rapid prototyping of sensing and analysis algorithms during feasibility testing.

The final component of our sensor-logging methodology is supporting a wide variety of diverse sensors. We use very general database relations for sensor logs and thus provide, in one unified output log, all sensors (real and virtual) that are necessary for Robot House studies. This allows us to capture, in feasibility and target studies, a wide set of phenomena for later analysis and ensures that data read during feasibility studies are representative of those that will be achieved during target studies, with respect to data ranges, precision, etc.

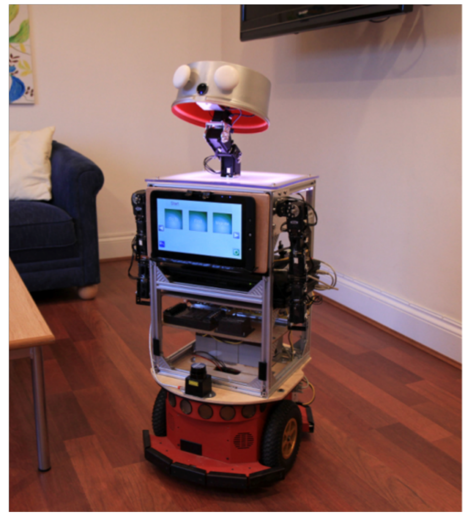

Figure 1. Sunflower robot at the UH Robot House.

1) Evaluation at the UH Robot House: At the UH Robot House, we utilize MySQL [11] as the database management system, which is free, open source, and has an existing ecosystem of software clients and tools. Alongside, two different but complementary commercially available sensor systems, the GEO System [12] and ZigBee Sensor Network [13], are currently installed in the Robot House. The GEO System is a real-time energy monitoring system for electrical devices. It is used to detect the activation and deactivation of electrical appliances by users, e.g. opening the refrigerator, turning on the kettle or the TV, etc. On the other hand, the ZigBee Sensor Network is used to detect those users' activities that cannot be identified by the GEO System such as opening of drawers or doors, occupied chairs or sofa seat places, opening of water taps, etc. In our case, the ZigBee Sensor Network consists of five ZigBee Wireless modules spread across the Robot House. They broadcast all sensors' changes through the Robot House ethernet infrastructure. Each module contains a different number of sensors connected, depending on its location and its use. Both sensory networks together offer a total of 59 sensors to be used in our HRI studies.

Most sensors were used in the Robot House before we introduced this logging method. Therefore, in order to centralize all data coming through both sensor networks, we developed a script connector responsible for updating every 1Hz all the sensors' changes into the aforementioned relational database. We also integrated virtual sensing of tablet touch events, which we will discuss later in the Robot Modeling section. In addition to sensor name and type, we also register location information, which allows us to visualize/analyze sensor logs according to smart-home regions. We have used this method in a single feasibility study with dozens of these sensors, tens of trials, over about an hour of total experimentation time. The default MySQL tuning parameters yielded performance that was sufficient for all of our sensors. During this period we have solidified the database schema and are currently investigating deployment across all studies in the Robot House. We are also investigating how this method can simplify and standardize logging and analysis tools, as well as

 ISBN: 978-1-61208-250-9

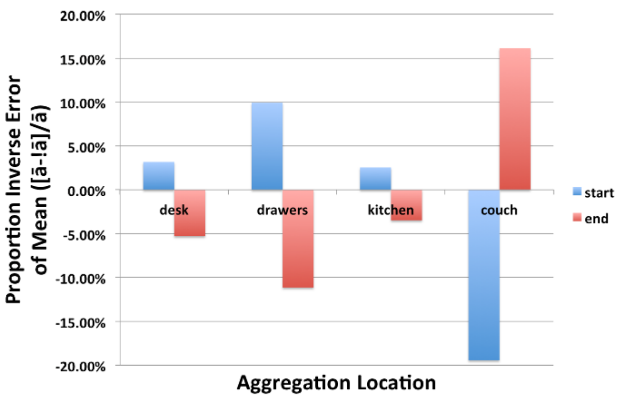

Figure 2. Analysis of navigation model error by location and directionality.

facilitate data sharing, of multiple robot models/smart houses in the ACCOMPANY project [14] across multiple countries/universities.

This method fully satisfies both requirements. Building on existing database technologies and software tools allows us to rapidly (R1) prototype studies as well as analyze experimental data across all sensors at our disposal. Adding a new, or modifying an existing sensor requires minimal time expenditure, allowing rapid prototyping. Also, as the method is ambivalent as to study type (feasibility vs. target) and goals, we can efficiently and seamlessly utilize our system across studies and all sensor logs reflect reported sensor values (R2).

B. Robot Modeling

In laboratories that pursue multiple ongoing HRI studies, robot time can become a scarce resource and time-sharing is required for robot hardware/software development and maintenance, human interaction trials, feasibility studies, etc. To address this issue, we have been investigating limited forms of robot modeling in feasibility studies that is human-robot interaction studies without the robot. While there can be obvious advantages to this approach in terms of resource efficiency (R1), the obvious concern is fidelity of any study outcomes: indeed, any form of simulation/modeling runs the risk of being "doomed to success" [15]. Furthermore, attempts to develop highly accurate, physics-based, noise-model-driven simulations can exacerbate this problem by being neither resource efficient (R1), nor sufficiently accurate for feasibility studies (R2).

Thus our research approach has been to explore the extreme of resource efficiency, tailored to each specific feasibility study. We ask first, what is the minimal role the robot plays in a particular target study and then we develop methods to minimally model this phenomenon, without a robot. Then, during feasibility testing, we act-out robotic interaction (discussed below) and log interaction events via virtual sensors.

1) Evaluation at the UH Robot House: The broader research goal of our target study was to investigate the role of robot long-term memory in HRI studies. Prior work (e.g. [16] [17]) has shown that long-term episodic memory has the potential to make robotic companions more capable and believable, but there are many open research issues, such as when/what to encode in memory, useful definitions of context for effective retrievals, as well as efficient and scalable algorithms for real-time retrievals over long periods of time [18]. In this context, our target study was to gather long-term episodic traces of HRI studies in the Robot House (on the order of weeks of trial data). Given this data set, one of research goals was to investigate general properties of traces (e.g. data-encoding rates, contextual patterns), as well as task-relevant learning opportunities (e.g. automatic context generation, user preference learning). For this specific goal, and the robot model/smart home, we recognized that we could perform feasibility studies given a small set of human-robot interactions in the context of a set of everyday tasks (e.g. cooking, cleaning, recreating).

We applied our robot-modeling method in two components. First, we investigated the fidelity of modeling two actions (robot navigation and opening/closing of a cargo drawer) of the Sunflower robot (see Figure 1) by observing a small set of trials, fitting a minimal parameterization, and assuming a Gaussian error model. Secondly, we investigated the effectiveness of having the feasibility participant manually log these interaction events via a tablet interface, while acting out the results of the interaction personally (e.g. personally transporting the logging tablet between locations, as opposed to the robot moving). At first glance, these approaches seem almost comically simple but that is the point. These approaches, as discussed below, are incredibly resource efficient, and we gathered data to assess tradeoffs in task-relevant fidelity.

Robot Operation Modeling: For each robot action (navigate-x-to-y, drawer-open/close), we performed three trials with the Sunflower robot in the Robot House and measured operation-completion time (precision = 1 second). For navigation, we used four strategically useful house locations and performed trials at all-pairs, gathering data independently for each direction (12 permutations x 3 trials = 36 data points). We recorded videos for all trials and the data for these models was collected and analyzed in under 1 hour.

Drawer operations executed with no measurable variance in time and symmetrically (i.e. no difference between the time to open and close). Thus, our model for robot drawer operations had zero parameters (2 seconds). We first hypothesized that we could usefully parameterize the navigation operation only by distance traveled. However, the data we gathered showed a poor correlation between the distance the robot traveled and the time required for navigation ($r^2 \approx 0.44$). In practice we could see, unsurprisingly, that more confined spaces caused the robot navigation algorithms to require additional time for correction. Since the locations were not uniform with respect to spatial constraint, we thus recognized the need to parameterize with respect to location.

Based upon this experience, our second hypothesis was that we could usefully parameterize the navigation operation by simply the two endpoints (un-ordered). However, the data showed that the types of navigation corrections necessary were dependent upon directionality: leaving a constrained area took much less time than entering that same area. Figure 2 illustrates our analysis that lead to this outcome, plotting the proportion inverse error of mean time for each location by direction ($[\bar{a}$ - $!\bar{a}]$ / $\bar{a}$, where $\bar{a}$ is average time and $!\bar{a}$ is average time between the same locations in opposite direction). We see in this data that not only is there an asymmetry per directionality in each location, but that asymmetry is not consistent between locations: for the desk, drawers, and kitchen, the error is lower for ending at that location as compared to starting, but this relationship is reversed for the couch (which has a table nearby). Consequently, our navigation model required start location, end location, and directionality.

Given these three parameters, three measurement trials yielded a relatively high variance (up to 35% of total operation time). For our feasibility study, this degree of accuracy was sufficient to gather relevant data. However, other studies may require tighter bounds, which may entail additional measurement trials and/or more sophisticated error models (leading to additional fidelity vs. resource-efficiency analysis).

Robot Interaction Modeling: To gather the data we required, it was necessary not only to know how long robot actions would take, but log these events in context of other Robot House sensor values while subjects were engaging in various scripted activities. This required (1) integrating robot event logs with other sensor inputs and (2) triggering these events at appropriate times without using an actual robot.

First, we registered a virtual sensor in our sensor-logging platform (see above). To trigger this sensor, we built an HTML5, touch-enabled web interface (see Figure 3) using the iUI framework [19]. The front-end interface (Figure 3: top) was arbitrarily extendable to support many robot actions and, upon submission, a PHP script would generate timing data based upon the robot operation model and send the data to our logging platform via HTTP. After logging the event, the interface displays confirmation and a countdown of operation time (Figure 3: bottom). Subtly, the physicality of the device used and this countdown provides feedback to the user of when a robot would be performing the action, and might thus be not available for interaction. The benefits of the technologies we used are rapid development; intuitive, touch interface for subjects; as well as platform- (Windows, *nix, Mac) and device- (desktop/laptop, tablet, phone) independence.

The source of the event trigger is a complex decision, and for preliminary evaluation we opted to evaluate the simplest choice: the subject would act-out the robot him/herself. Thus, if the subject wished for the robot to take cargo (e.g. a plate) to a destination (e.g. the kitchen), s/he would perform the following sequence of actions: log a ”drawer open” action (wait for countdown), log a ”drawer close” action (wait for countdown), log a ”navigate” action, bring the interface device and cargo to the destination, leaving both (and not having further interaction with the interface till countdown completed). As expected, this approach places numerous cognitive and physical burdens on the subject, such as remembering to initiate robot actions [in proper order], maintaining task priorities, and actually performing actions. This approach also affects fidelity (R2) in that the subject cannot simultaneously perform a robot action and a study action. However, the resource cost of the feasibility study is now limited to a single individual and no robot (R1). In our feasibility-study trials, we found that even a well-trained subject was likely to make numerous event-logging errors (e.g. forgetting to log an event), and thus in the future we plan to evaluate a slightly more complex act-out option: utilizing a second person as an independent actor (sacrificing R1 for R2).

C. Interaction Script Generation

In designing long-term HRI studies, a great deal of time can go into developing user-behavior scripts. Some of the complexities

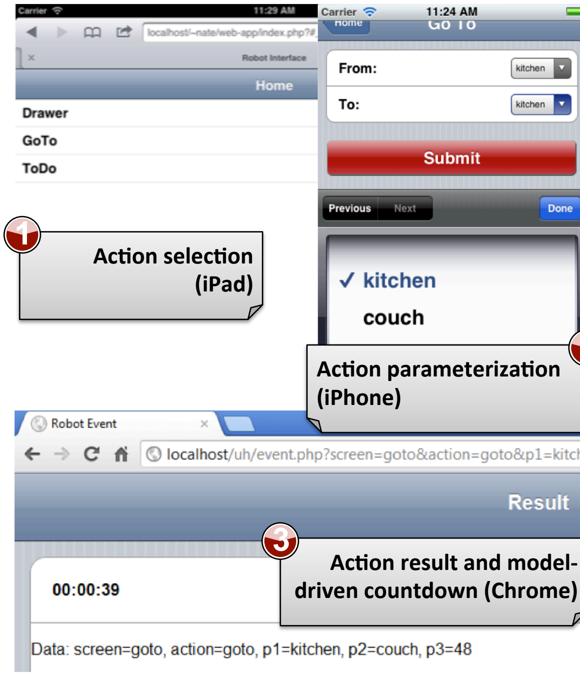

Figure 3. Robot-interaction and virtual-sensor interface (sequence of screenshots for an action on three different devices/programs).

involved are reflecting the appropriate level of abstraction in subject tasks (reflecting necessary task structure, while not inhibiting how subjects individualize their execution); controlling for sources of stochasticity in task organization and execution; as well as scaling to large numbers of participants over long periods of time. We believe that feedback from rapid feasibility studies can improve the overall time to develop user-behavior scripts, and have focused on a flexible, intuitive framework to describe user activities and sources of randomness, as well as quickly and reproducibly produce scripts according to this framework.

Before we describe our method, we note that the problem of producing user-behavior scripts has many elements in common with controlling non-playing-character (NPC) behavior in video games (e.g. [20] [21]), assuming individual users’ behavior is driven by simplified motivational states. For simple NPCs, finite state machines (FSM) can fully describe behavior. However, for more complex characters (or units of characters) in interesting environments, it becomes too complex to develop and maintain a step-by-step characterization of every suitable action in every state (especially for non-programmers) [21]. However, on the opposite extreme, it is often too computationally expensive to simply utilize off-the-shelf planners during real-time play [20]. Thus, one reasonable approach to this problem is to have an intuitive, often graphical framework by which to hierarchically describe goals and organizations of contexts (e.g. via Hierarchical Task Networks, or HTNs); efficiently instantiate a script based upon this organization; and then rely upon computationally efficient algorithms to implement script primitives. At a high level, this describes our method for user-behavior script generation.

Our hierarchical description language draws inspiration from HTNs: an acyclic graph where internal nodes are used for or-

ISBN: 978-1-61208-250-9

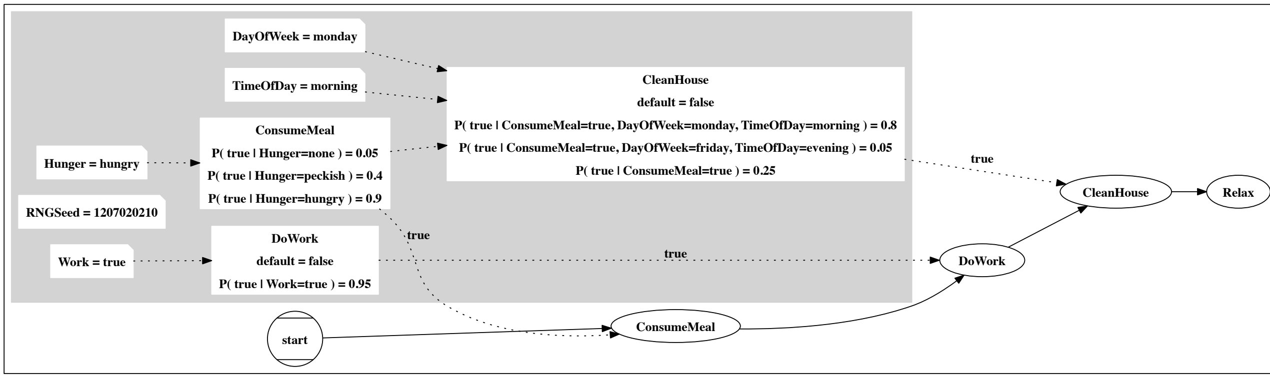

Figure 4. Example top-level script-structure description.

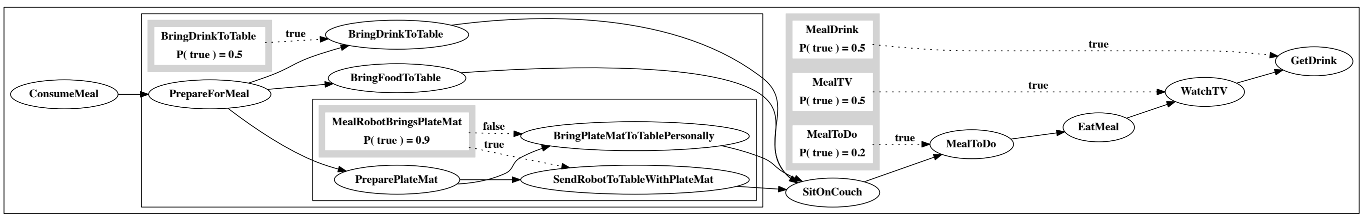

Figure 5. Expansion of Figure 4, showing substructure of the ConsumeMeal node.

ganization and leaf nodes indicate behavior primitives. However, drawing on Bayesian models for inspiration, we also support describing probabilistic dependencies. Thus, each internal node supports an arbitrary set of variables, and each edge supports a conditional-probability table (CPT), describing uniform distributions over actions. For example, consider Figure 4, an example used at the UH Robot House that describes a high-level sequence of user behaviors consisting of consuming a meal, doing some work, cleaning the house, and relaxing. The greyed area shows variables, variable dependencies, variable values (for a particular script instantiation), and conditional-probability distributions of performing actions based upon variable values. For example, there is a 90% chance that the subject will be instructed to consume a meal (and, if so, an 80% chance that the subject will be instructed to clean the house). Each of the nodes supports arbitrarily deep recursive structure, depending upon the needs of the study (e.g. see a possible expansion of the ConsumeMeal node in Figure 5). Given this high-level description, it is possible to efficiently generate an arbitrarily large set of output scripts that abide by the structure (by controlling input variables and random-seed generation).

1) Evaluation at the UH Robot House: We implemented the high-level description language as a Java library. To generate scripts we implemented a hierarchical-decision agent using the Soar cognitive architecture [22] [23]: the Java library compiles the graph, inputs it to the agent (along with requisite variable values and, optionally a random seed to control stochastic decisions), and receives a sequence of leaf nodes as a script-agent output. We also implemented visualization routines using Graphviz [24] for debugging and communication purposes (e.g. see Figures 4 and 5).

In our feasibility studies, we wrapped the above library/agent within a client program that requested as input all information required to uniquely identify a subject trial (e.g. date, trial #) such that the output script could be later regenerated for examination or as input to learning/activity-recognition algorithms. The script generation program takes a fraction of a second to run for a graph with dozens of nodes and variables. For example, here is a sample input/output based upon the graphs described in Figures 4 and 5:

```
Input> year=2012 month=7 day=5
       time=morning hunger=peckish
       work=true trial=3

Output> start ->
        SendRobotToTableWithPlateMat ->
        BringFoodToTable ->
        BringDrinkToTable -> SitOnCouch ->
        EatMeal -> GetDrink ->
        TurnOnMusic -> WorkToDo ->
        WorkForTenMinutes ->
        GetPeriodicalFromDrawers ->
        LieOnCouch -> RelaxToDo -> Read
```

This example illustrates how an intuitive, hierarchical description of a complex, stochastic (but structured) study can efficiently result in a script that has various degrees of specificity. For example, where necessary there are very concrete actions (e.g. sending the robot to the table with the plate mat), but also those with a great deal of flexibility left to the subject (e.g. bring food to the table). In one feasibility study, we used this software to generate a suite of plans (3 parameter settings, 2 trials each, for 1 subject), all of which in about 1 minute of time (R1).

To facilitate fast and accurate (R2) development, in the future we plan to develop a GUI to support drawing of graph nodes and visually entering CPT values for non-programmers.

IV. CONCLUSIONS

In this work, we have described and evaluated methods to perform HRI feasibility studies that can dramatically reduce resource utilization when compared to full target studies, but maintaining the high level of HRI fidelity. We have incorporated all of these methods in our work at the UH Robot House. Our sensor logging approach has clearly demonstrated effectiveness and we are beginning deployment across all of our studies, while the other methods require additional investigation.

The goal of this paper was to begin a conversation in the HRI community around the value of feasibility studies for complex HRI scenarios and the need to investigate and disseminate software and lessons-learned from practical techniques therein. We hope that future work highlights innovations in this space, which we contend will improve the speed and quality with which studies can be executed in this field.

ACKNOWLEDGMENT

The first author thanks the National Science Foundation for funding an extended research visit to the UH Robot House, without which this research would have not been possible. The work described in this paper was partially supported by the EU Integrated Project ACCOMPANY (Acceptable robotiCs COMPanions for AgeiNg Years) [14]. Funded by the European Commission under contract number 287624.

REFERENCES

[1] Michael L. Walters, Sarah Woods, Kheng Lee Koay, and Kerstin Dautenhahn. Practical and methodological challenges in designing and conducting interaction studies with human subjects. In *University of Hertfordshire, UK*, pages 110–120, 2005.

[2] Kertin Dautenhahn. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):679–704, 2007.

[3] Kerstin Dautenhahn. Methodology and Themes of Human-Robot Interaction: A Growing Research Field . *International Journal of Advanced Robotic Systems*, 4(1):103–108, 2007.

[4] University of Hertfordshire (UH) Robot House. http://www.hertsrobots.com/. [Online; accessed 12.12.2012].

[5] Christoph Bartneck and Jun Hu. Rapid prototyping for interactive robots. In *The 8th Conference on Intelligent Autonomous Systems (IAS-8)*, pages 136–145. IOS press, 2004.

[6] Michael L. Walters, Manja Lohse, Marc Hanheide, Britta Wrede, Kheng Lee Koay, Dag Sverre Syrdal, Anders Green, Helge Hüttenrauch, Kerstin Dautenhahn, Gerhard Sagerer, and Kerstin Severinson-Eklundh. Evaluating the robot personality and verbal behaviour of domestic robots using video-based studies. *Advanced Robotics*, 25:2233 – 2254, December 2011.

[7] Amiy R. Chatley, Kerstin Dautenhahn, Mick L. Walters, Dag S. Syrdal, and Bruce Christianson. Theatre as a discussion tool in human-robot interaction experiments - a pilot study. In *Proceedings of the 2010 Third International Conference on Advances in Computer-Human Interactions*, ACHI '10, pages 73–78, Washington, DC, USA, 2010. IEEE Computer Society.

[8] Ben Robins, Kerstin Dautenhahn, and Janek Dubowski. Does appearance matter in the interaction of children with autism with a humanoid robot? *Interaction Studies*, 7(3):509–542, 2006.

[9] Anders Green, Helge Hüttenrauch, and Kerstin S. Eklundh. Applying the wizard-of-oz framework to cooperative service discovery and configuration. In *Proceeding of the 13th IEEE International Workshop on Robot and Human Interactive Communication, ROMAN04*, pages 575–580, Kurashiki, Okayama, Japan, September 2004.

[10] Andy Clark. *Being There: Putting Brain, Body, and World Together Again*. The MIT Press, January 1998.

[11] MySQL. http://www.mysql.com/. [Online; accessed 12.12.2012].

[12] GEO: Green Energy Options. http://www.greenenergyoptions.co.uk/. [Online; accessed 12.12.2012].

[13] Shahin Farahani. *ZigBee Wireless Networks and Transceivers*. Newnes, Newton, MA, USA, 2008.

[14] ACCOMPANY: Acceptable robotiCs COMPanions for AgeiNg Years. Project No. 287624. http://www.accompanyproject.eu/, October 2011. [Online; accessed 12.12.2012].

[15] Rodney A. Brooks and Maja J. Mataric. Real robots, real learning problems. In Jonathan H. Connell and Sridhar Mahadevan, editors, *Robot Learning*, volume 233 of *The Kluwer International Series in Engineering and Computer Science*, pages 193–213. Springer US, 1993.

[16] Paulo F. Gomes, Carlos Martinho, and Ana Paiva. I've been here before! location and appraisal in memory. In *Proc. of The 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*. IFAAMAS (http://www.aamas-conference.org/), 2011.

[17] Mei Yii Lim, Ruth Aylett, Patricia A. Vargas, Wan Ching Ho, and João Dias. Human-like memory retrieval mechanisms for social companions. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '11, pages 1117–1118, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[18] John E. Laird and Nate Derbinsky. A year of episodic memory. In *Proceedings of the Workshop on Grand Challenges for Reasoning from Experiences*, pages 7–10, Pasadena, CA, USA, July 2009.

[19] iUI. http://www.iui-js.org/. [Online; accessed 12.12.2012].

[20] John-Paul Kelly, Adi Botea, and Sven Koenig. Offline planning with hierarchical task networks in video games. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Interactive Digital Entertainment AIIDE-08*, pages 60–65, 2008.

[21] Antonio A. Sánchez-Ruiz, David Llansó, Marco A. Gómez-Martín, and Pedro A. González-Calero. Authoring behaviours for game characters reusing automatically generated abstract cases. In Sarah Jane Delany, editor, *ICCBR Workshop*, pages 129–137, July 2009.

[22] Soar v9.3.2. http://sitemaker.umich.edu/soar/. [Online; accessed 12.12.2012].

[23] John E. Laird. *The Soar Cognitive Architecture*. MIT Press, 2012.

[24] Graphviz. http://graphviz.org/. [Online; accessed 12.12.2012].