# The Making of a Dataset for Smart Spaces

Eunju Kim, Sumi Helal, Jae woong Lee and Shantonu Hossain

Mobile and Pervasive Computing Laboratory.
The Department of Computer and Information Science and Engineering
University of Florida
{ejkim, helal, jwlee, shossain}@cise.ufl.edu

**Abstract.** In this paper we propose a two-phase methodology for designing datasets that can be used to test and evaluate activity recognition algorithms. The trade offs between time, cost and recognition performance is one challenge. The effectiveness of a dataset, which contrasts the incremental performance gain with the increase in time, efforts, and number and cost of sensors is another challenging area that is often overlooked. Our proposed methodology is iterative and adaptive and addresses issues of sensor use modality and its effect on overall performance. We present our methodology and provide an assessment for its effectiveness using both a simulation model and a real world deployment.

**Keywords:** Activity Dataset, Activity Dataset Design, Activity Recognition, Pervasive Space Simulation.

## 1 Introduction

Activity recognition technology is critical to many human-centric ubiquitous applications. Activity models, activity recognition algorithms, and activity recognition sensor platforms are active areas of research [1][4][5][6][8][10]. Experiments are necessary for validating novel ideas and collecting comprehensive datasets. Furthermore, a variety of experimental setups are required due to the diversity in human activities. For instance, depending on the number of residents or activity order such as sequential or concurrent activity, different sensor setup will be necessary. However, building an experimental environment that closely mimics real-world applications requires significant effort and cost. Therefore, it is difficult to build necessary experimental environments. As a solution, many research groups have shared their activity datasets [4][5][6][7][10]. However, since sensor technology is rapidly evolving, researchers often need to upgrade or replace existing activity dataset. This demands new methods to create activity data. Research on activity dataset has mainly focused on classifying the collected data into different types [4][5][6][7]. But, so far there has been little work done on effective generation of activity dataset at low cost, effort and time. In this paper, we will propose a methodology for creating effective activity datasets with lesser cost, time and errors as compared to existing methods.

## 1.1 Motivation

Our motivation is developing processes and tools to create a highly effective activity datasets. We observe that activity datasets, even in the same domain, may considerably differ due to the number of sensors, sensor types, and the way the sensors are used (sensor use modality). This diversity of datasets in the same domain raises several issues. In other words, when we verify new activity recognition algorithm or activity model, the experimental results may be different depending on the activity dataset.

First of all, using more sensors may trade-off cost for achieving higher activity recognition performance. For example, if we use many sensors, it may be better to detect an activity. However, it is also more expensive. Therefore, it is necessary to find an effective number of sensors.

Secondly, finding the proper type of sensors is crucial to generate an effective dataset. For instance, although a sound sensor can be used to recognize eating activity, other type of sensors such as RFID or camera sensors may produce more accurate data when people watch television while eating food.

Lastly, the effectiveness of a dataset can be achieved by a modality analysis of sensors such as install location (wearable, environment), size, and other constraints. In other words, sensor should be carefully fitted (and sometimes, conditioned) to best suite the intended purpose. For example, if we plan to install pressure sensors on a bed to detect a sleeping activity, we should carefully analyze and find the most appropriate location on bed and we should adjust the operational range of the sensor to better detect the weight of a person (e.g., use appropriate resistor to shift range of sensitivity).

As shown above, it is important to determine the effective number of sensors, type, and usage modality to accurately detect the target activity and to collect high quality corresponding datasets. Therefore, it is essential to develop a new methodology which allows discovering the most serious erroneous flows during the design phase, leaving only little errors and adjustments to be dealt with before actual deployment.

## 1.2 Proposed Approach

As a solution to the aforementioned problem, we propose a two phase activity dataset design approach. Our approach is composed of three steps: activity design, simulation, real-world instrumentation. Instead of building a real sensor setup immediately after activity design, we have a simulation step in the middle of the process. This approach has several advantages. First of all, it can help in discovering and selecting a most effective sensor set for a specific purpose. In many cases, the sensor setup relies on the knowledge and intuition of the researcher. If too few or too many sensors are used, the quality and effectiveness of the dataset could be questioned. By performing simulation, this risk will be reduced. Secondly, our approach can save time and cost since all investigations are performed in a tightly controlled environment. Finally, the activity dataset produced by our approach can be utilized as a standard dataset to verify the activity model under investigation along with its activity recognition algorithms.

The rest of this paper is organized as follows. In section 2, we will describe several related works. The proposed approach is explained in section 3. The experiments and results are discussed in section 4. Finally, section 5 concludes the paper.

## 2 Related Work

There have been many approaches to make dataset for activity recognition. In this section, we will talk about experimentally obtained activity recognition dataset in real world and simulation based activity dataset.

### 2.1 Real-World Activity Recognition Datasets

Activity recognition datasets have been collected for archiving by several research groups. MIT provides three datasets called PlaceLab Intensive Activity Dataset1 (PLIA1), PlaceLab Intensive Activity Dataset2 (PLIA2), and PlaceLab Couple Dataset (PL Couple) [4][5][6]. In PLIA1, the researchers built a real experimental environment in a 1000 sq. ft. apartment and installed approximately 214 sensors such as temperature, humidity light, pressure, current, water flow, gas flow, object, accelerometer, camera, and microphone sensors [4]. They performed a set of common household activities during 4-hour period and collected sensor data of 89 activities [4]. In PLIA2, they introduced a portable wireless sensor platform named MITes [5]. It is useful to collect human activity data in real environment such as home [5]. The MITes platform includes five types of wearable sensors such as accelerometers, heart rate, ultra violet radiation exposure, an RFID reader in a wristband form factor, and location beacons [5]. It also includes six environmental sensor types such as light, temperature, switch, object, proximity, and current [5]. This sample PlaceLab dataset2 was recorded for four hours with a volunteer who was familiar with the PlaceLab, but not a creator of the core technical infrastructure. He performed a set of common household activities during the four-hour period using a set of instructions [5]. This dataset is provided with a visualization tool [5]. The third dataset, PLCouple1, consisted of 100 hours of annotated data from a couple who lived in the PlaceLab [6]. This experiment is performed for 15 days with 900 sensor inputs including wired sensors, motion-detection sensors, and RFID tag. The purpose of this experiment is to compare different sensor modalities on data in real environment [6].

Artificial Intelligence Laboratory, Washington State University provides several activity datasets for single resident, two residents, and multiple-residents. The datasets are further specialized into normal activity, abnormal activity, separate activities, interwoven activities, and smart home/smart workplace datasets, respectively [7][8]. Human subjects lived in the laboratory-built smart apartment and the smart workplace as part of the CASAS project [7]. The smart apartment had three bedrooms, one bathroom, a kitchen, and a living/dining room. Temperature sensor and custom-built analog sensors were installed to monitor motion, water, burner, telephone, and item use. These sensors were used to detect daily living activities [8]. The smart workplace is a lab which is organized into four cubicles and a server area, a postdoc office, a meeting room, a lounge, and a kitchen. Every cubicle had a desk and chairs, and a computer. Motion sensor, power line controllers, and magnetic open/close sensors are installed in the space. This is used to recognize social interaction activities among people in the lab such as apart, coming, going, or joint together [7] [8]. These dataset also provided a visualization tool so users can utilize it for annotation of the sensor dataset [7] [8].

University of Amsterdam provides a real world activities of daily living dataset [10] (we will refer to it in this paper as the Amsterdam dataset)   This data set records ADL performed by a 26-year-old man living in a three-bedroom apartment for 28 days [10]. The researchers installed 14 state-change sensors in the house. Data has been recorded for only 24 days out of which only 23 days have annotations. Sensors are installed in several places in the apartment including doors, cup-boards, refrigerator, and toilet flush. Activities (such as 'Leaving', 'Toileting', 'Showering', 'Sleeping', 'Drink', 'Breakfast', and 'Dinner') are annotated by the subject himself using a Bluetooth headset. Since PLIA1 and Amsterdam dataset provide text format annotation files, they are best suited for activity recognition research [10]. On the other hands, PLIA2 and Washington State University datasets provide visualization tools, they are more suitable to detect activity patterns or activity episode discovery studies [4][5][6][7] [8].

The main focus of collecting datasets has been to closely represent actual activities in real environments. For example, the PLIA dataset tried to overcome the limitations imposed by laboratory environment [4]. Washington State University dataset collected multiple residents and interwoven the dataset because they are closer to the real-world situation [8]. In the Amsterdam dataset, data was collected about a volunteer for 28 days. Therefore, these datasets are a step toward realizing real-world applications [10]. Our research takes activity recognition a step forward by generating effective datasets. Collecting activity datasets requires not only high cost but also time and effort. Datasets having the same domain may differ due to different sensor types and numbers. Some sensors may be redundant while in some cases the number of sensors may be insufficient.

### 2.2 Simulation based Activity Datasets

Generating meaningful sensory data is one of the major impediments in human activity recognition research. Researchers often need data to evaluate the viability of their models and algorithms. But useful sensory data from real world deployments of pervasive spaces is very scarce. This is due to the significant cost, and elaborate groundwork needed to create actual spaces. [11]

Given the aforementioned challenges, simulation is a promising and sensible alternative in practical ways to experiment with human activities in pervasive spaces. Powerful and realistic simulation tools could be used to support the growing demand for test data. Simulation enables researchers to create focused synthetic replications of important events and activities under study. It can be easily changed and refined allowing the researchers to experiment, analyze and fine-tune their models and associated algorithms. Simulation also allows a wider community of researchers to engage and collaborate to solve a specific problem. Hence, a design based on preliminary simulation studies would most likely to be a more robust and inclusive design. Also, a simulation model that mimics an existing real world is most likely to answer more questions (generate much more data) than the actual space. This early stage simulation can help researchers evaluate their ideas and algorithms quickly and with reasonable accuracy. [12]

There have been several approaches to introduce the simulation concepts and algorithms and to develop the simulating tools. SENSORIA [14], is a simulator focusing on traffic generation, energy consumption and inherent protocols of wireless sensor network (WSN). In [20], a detailed simulation model was presented which also focuses on accurate model for battery, processor power consumption and in network traffic. In [17], Discrete-Event system Specification (DEVS) was proposed to define asynchronous discrete-events to be simulated. Since these approaches suggest modeling simulation events occurring in WSN environment, routing and communication becomes non-trivial factors for them. However, the events and environments they are modeling could not be available to simulate human activities. They have lacks of specific and complex labeling which is necessary to describe human activities performed in pervasive environment.

DiaSim [16] simulator executes pervasive computing applications by creating an emulation layer and developing simulation logic using a programming framework. It describes a pervasive computing environment in terms of stimulus procedures (any change in the environment that can be consumed by sensors) and simulated services (sensors and actuators) in a specification language called DiaSpec. However, DiaSim simulates applications such as fire situations, intrusions, etc. to identify potential conflicts. It does not care much for human activities performed in pervasive spaces.

The human activities can be simulated through Persim [11], [18] developed at University of Florida. It allows researchers to fine-tune simulation models until his/her satisfaction. However, there is still a major roadblock that impedes the simulation through Persim which is: How do they know that a designed model is reasonable enough to simulate dataset closed to dataset generated in the real space? Because costs in establishing a virtual pervasive space and simulating it are very low, researchers can design what they want to establish. But these models and designs might not be realistic in terms of possibility that the designed spaces could be existed. In order to get much more reasonable yet still accurate simulation mode and its dataset, this paper proposes a heuristic way to design an effective space through fine-tuning experiments.

## 3 Two Phase Dataset Design Approach

We developed a two phase approach for activity dataset design as shown in Fig. 1. Phase 1 is concerned with generating activity dataset using simulation after detailed activity design. The purpose of the detailed activity design is looking at the nature of human activities and matching the activities with the most suitable sensor technology, and finding the specific use modalities that boosts the effectiveness and the contributions of the sensor to the target activity.. The detailed activity design is followed by generation of activity dataset through simulation. Phase 1 goes through multiple iterations until the target dataset achieves high recognition performance. Once this is accomplished, we move to phase 2 in which we implement the actual sensor instrumentation and collect actual activity data in the real sensor environment. This real world activity dataset would be more effective and accurate compared to that of a single phase activity dataset.

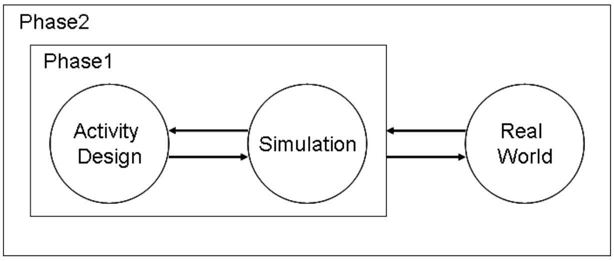



**Fig. 1.** Two phases activity dataset collection process.

### 3.1 Activity Design

Designing activities is a detailed step in the making of a dataset. It consists of four sub-steps: Generic Activity Model Design, Selection of Target Activities, Functional Instrumentation, and Modality Analysis. We describe each of these sub-steps in the following sub-sections.

#### 3.1.1 Generic Activity Model Design

A well-defined activity model is very important because other steps are directly influenced (limited or empowered) by the activity model. In [1], we proposed a generic multi-layer activity model that is shown in Fig. 2. The generic model was intended to provide clear separation between the sensors layer (observation sub-system) and the rest of the activity model (any model). The goal of this separation is to enable a more scalable approach to try and error research in which researchers are empowered to explore varieties of sensors until they arrive at he most "effective" sensor set. In this paper, we instantiate this generic model based on the activity design on hand. A brief description of each layer of the generic model is shown below.

**Sensor data.** This is the data from installed sensors in the pervasive space (e.g. smart home). Based on the source of data, sensor data is classified into four types: motion, tool, object, and context sensor data. Motion sensor data is about peoples' movements such as raising an arm, turning body, or folding legs. Tool sensor data come from sensors attached to the objects which are used by the people. For example, spoon or

fork is a tool for eating. Object sensor data is from sensors installed on passive objects such as grocery or frozen food packets. [1].
**Action.** Action is a unit behavior, which can be directly determined by combination of tool and motion sensors. For example, an eating activity is composed of several actions such as scooping, picking or cutting food [1].

**Activity.** Activity is a collection of combination of actions and objects. Activity may involve multiple actions that occur in a certain order. But because the order of actions varies a lot according to the user, we consider the relationship between activity, action and object in defining activities [1].

**Meta activity.** A meta activity is a collection of related activities. It is a more abstract entity than an activity [1].

**Context.** Context is information, which is used to determine a situation [1]. Contexts are classified into directly sensed, directly defined, and indirect context [1]. A directly sensed context is a record of user activity or sensor data. As the indirect context reflects the inclination of a user, it needs to be found by a system and activity history. A directly defined context is predefined because it is common knowledge. A directly defined context is presented to a case and the case is a pair of problem and solution or cause and effect [1].

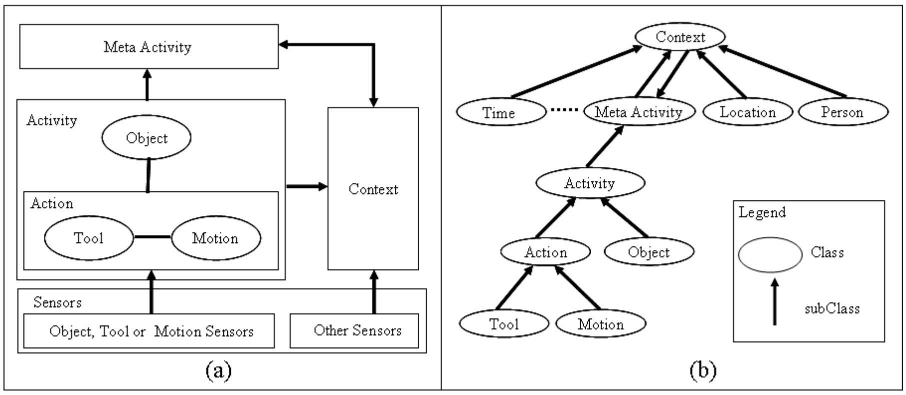



**Fig. 2.** Generic Activity Model, (a) block diagram of activity model (b) class hierarchy for activity ontology.

### 3.1.2 Selection of the Target Activity

We need to decide the domain of activities. Activity domain is classified in many ways such as home, office, school, gym, or kindergarten. For example, if we want to collect daily living activities at a home of an elderly person, then there would be several activities such as sleeping, eating, cooking, and etc [3]. Among all activities, we need to choose a subset and define the meta activities, activities, and actions according to our generic activity model.

#### 3.1.3 Functional Instrumentation

Once we decide the target activities, we need to find which information needs to be sensed. For example, for "eating" activity, there are many types of information such as eating sound, eating motion, and the amount of food consumed, food smell, or location of user. However, collecting all those information may not be necessary. Therefore; we should choose an effective combination of this information. After we determine the information to sense, we need to decide proper sensors according to its function. For instance, there are acoustic sensors for detecting sound and several types of motion sensors for sensing human motions. However, there will always be information that no available sensor can support. For instance, detecting nutrition information the elderly take is very helpful in tracking their health status. However, it may not be easy to find suitable sensors that can sense such nutrition information. Therefore, only effective and synergistic sets of information and sensors are chosen in this step.

#### 3.1.4 Modality Analysis

Once we decide upon the information to be detected and sensors to be used, we should find how to utilize sensors to acquire information most effectively. This is a question of modality of use of the sensor. Modalities include the number of sensors to use, the way a sensor is worn or the location it is placed at, the feasibility of sensors such as size or performance. All such relevant modalities are analyzed in this step. For example, if eating is a target activity and if we want to detect eating motion, motion sensor will be chosen. If there is a very small motion sensor, eating activity can be detected by the motion sensors installed in spoon or fork. Otherwise, residents may need to wear motion sensor on their arm or hand, which is obviously a great inconvenience and therefore must be avoided. Also, the spoon and fork will need to be washed. So the sensors should be durable to withstand a wash cycle. Otherwise, alternative ways to detect the same activities must be explored. Another example is an RFID reader. RFID readers are classified according to their frequency band and reading (sensing) range. If their reading range is too small, say a few centimeters, then it will be difficult to detect important information (tags) farther than the range. On the other hand, if the reading range is too large (e.g. 20 meters), then unrelated information may unnecessarily get sensed which may cause confusion or errors. Hence, paying attention to the specific type of reader and tags is of paramount importance. Modality analysis is indeed very important in addressing all above issues.

#### 3.1.5 Sample Design

We designed target activities based on a scenario in which Mrs. Smith, an 87 years old woman, lives alone in a smart house [3]. A detailed analysis of her daily life reveals 11 meta activities and 31 activities which we casted into our generic activity model as shown in Table 1.

**Table 1.** Meta activities, Activities, and Actions of Mrs. Smith's scenario

| Meta Activity | Activity | Action |
|---|---|---|
| Rest | Sleeping, Relaxing | Lying down, Getup |
| Having a meal | Eating, Drinking | Cutting, Picking, Scooping, Serving, Lifting a cup |
| Getting out | Leaving & Arriving home | Open front door, Close front door, Standing on a door mat. |

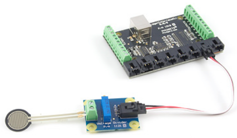
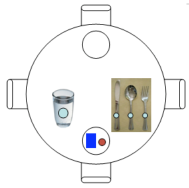
In the Functional Instrumentation step, we decided on information and sensors consistent with Table 1. The information for *sleeping* or *relaxing* activity recognition is pressure and vibration level of bed or sofa. For eating and drinking activity, usage of *eating* and *drinking* tools such as spoon, fork, knife, plate, or cup is chosen. To recognize the leaving and arriving activity, door status or the pressure of door mats (on both sides) will be used. Table 2 shows an example of the specific outcome of this step for *sleeping* and *eating* activity.

**Table 2.** Example of Activity and Sensor Designs for *sleeping* and *eating* activities

| MetaActivity<br>- Activity<br>(Location) | Actions | Information & Modality<br>(Possible place to install sensors) | Sensor (number of sensors) |
|---|---|---|---|
| Rest<br>- Sleeping<br>(Bedroom) | • Lying-down<br>• Getup | • Pressure of mattress<br>- Below a mattress<br>• Vibration of mattress<br>- Top of a mattress | • Vibration sensor (1)<br>-Phidgets 1104 Vibration Sensor [9]<br>• Force sensor sensor (1)<br>-Phidgets 3102 8" FlexiForce sensor 0-100lb. Resistive Force sensor [9] |
| Having a meal<br>- Eating<br>(Dining room) | • Cutting<br>• Picking<br>• Scooping | • Usage of eating tools<br>- Dish or Bowl<br>- Spoon<br>- Fork<br>- Knife | • RFID Reader (1)<br>-Phidgets 1023 RFID reader [9]<br>• RFID Tag (2)<br>- Phidgets 3007 RFID tags [9] |
| | • Serving | • Weight of food<br>- Below a dining mat | • Pressure sensor<br>-Phidgets 3105 Interlink Electronics 1.5" Square FSR [9] |

In modality analysis, we analyzed the sensor technology available in the market and choose the most synergetic sensors that fit the target activities. Table 3 presents an example of sensor deployment for eating activity based on the modality analysis.

**Table 3.** Example of Modality Analysis for *eat*ing activity

| Activities | Infrastructure Requirement | |
|---|---|---|
| Eating | • Integration Equipments<br><br>- Phidget 1070 [9], Phidget SBC [9] or Portable device for networking (eg. Smart phone)<br><br>• Integration Schema<br> | Modality Justification:<br><br>GTSH Dining room<br> |

| Symbol | Sensor | Conditioning |
|---|---|---|
| ■ | RFID reader | No |
| ○ | RFID tag | Yes |
| ● | Force | Yes |

## 3.2 Activity Simulation

Now we move into the simulation step in Phase 1. The Persim simulator [15], which is an event-driven, human activity simulator was used. Persim is capable of capturing the physical elements of a space including its sensors, actuators and human activities. Persim users typically build a simulation "project" over multiple sessions before they are ready to generate data or make a multitude of changes to the sensory elements, the activities or even the structure of the output dataset. Data generated by Persim follows the Sensory Dataset Description Language (SDDL) proposed standard. Persim promotes sharing of efforts through the use of standardized dataset representation (SDDL) format standard [13][19]. This allows one user to start a simulation project by uploading a dataset originally generated by another user (through Persim), modifying the design and fine-tuning the experiments to achieve a specific research-goal.

One powerful feature of Persim is its ability to weave simulated events into actual events in datasets represented in SDDL. This feature empowers the owner or any other user of an actual dataset to go back in time and explore slight variation in the actual space without actually repeating the experiments or collecting additional sensor data. Thus the simulator is intended to open a new dimension of collaborative research in the area of human activity recognition and other simulator applications.

### 3.2.1 Scenario-based simulation

In order to simulate activities, Persim needs a scenario which describes the target activities. The scenario includes where the activities happen and how they occur. Hence how well and how realistic the scenario is defined is the main key in obtaining meaningful simulation results. Daily activities in the morning could be one of the examples of scenarios. In the morning a human usually gets up, jogs, and then has breakfast. These activities occur in the bedroom, outside of the house and in the dining room. The sensors and RFIDs which are deployed or attached on objects in the areas generate data when they detect moves related to the activities.

### 3.2.2 Mapping Scenario to Simulation

There are four essential steps to map scenario to simulation. First, a space and its various areas are designed to create higher level of realism through the use of space templates such as single family home, apartment, etc. In the second step, sensor/actuator components provide sensors and actuators to be deployed in the designed space. Following the scenario, sensors and actuators are located at proper positions with the correct attributes such as the type of sensor, sensor event generator, and domain value generator. Note that these attributes also can be edited to get fine result of simulation. Thirdly, activities and actuation rules are added. Activities rules are defined with their own name and include behaviors such as walking from an area to another area or object interactions such as grabbing a spoon. Actuation rules specify the logic of the actuation based on sensor events and in terms of invoking actuator(s).

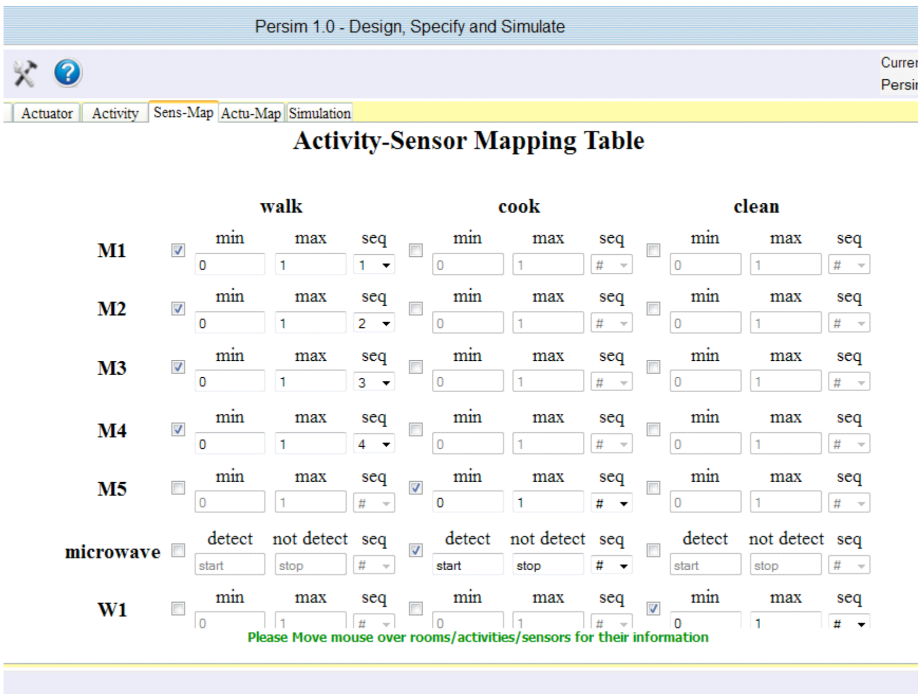



**Fig. 3.** Activity-sensor mapping table.

In the fourth step, the user prescribes two important mappings: activities-to-sensors mapping. The former mapping specifies which sensors are relevant to the detection of each activity. For easy and convenient mapping, Persim provides a mapping table, shown Fig. 3. In the latter mapping, each actuator is mapped twice, once to the set of sensors that could trigger it, and once again by the set of sensors that could be affected by that actuator when triggered. Persim also has a mapping table for sensors and actuators.

In the last step, a user finalizes simulation configuration with proper parameters such as simulation time and process generating intervals. Fig. 4 shows the simulation configuration table. Finally, Persim is ready to simulate activities as defined in the scenario. Note that the user may repeatedly go over these five steps to make changes or fine-tune the simulation.

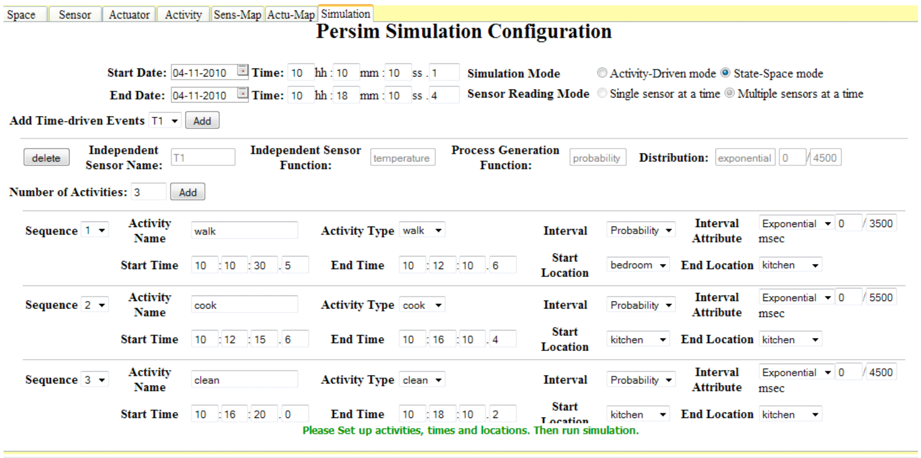



**Fig. 4.** Persim simulation configuration.

# 4 Experiments and Results

We validated our two phase dataset approach in terms of effectiveness and accuracy of the generated dataset through the experimentations. We performed both simulation and real sensor deployment/data collection. First, we designed a simulation model (e.g. parameters) of two designed activities. Next, we simulated the two activities with respect to the model and generated datasets. Then, we instrumented a space with real sensor configuration and collected sensor data of the performed activity. We compared the effectiveness and performance of the two datasets using an activity recognition algorithm/system. The premise of the experiment is that if the accuracy of the simulation dataset are similar to the real sensor dataset and their effectiveness are the same, we conclude that the two-phase dataset design approach is useful. To illustrate, in terms of effectiveness, both simulation and real sensor dataset should have the same result because simulation should help to find the most effective sensor set. However, the accuracy of simulation dataset can be higher than real sensor dataset because certain unexpected situations that occur in real-world are not accounted for in the simulation model. For example, if we want to collect sleeping activity data for 20 minutes, simulation data will include sleeping activity data for this time. But real sensor dataset can miss some part of the activity data due to sensor noise and network delay. Section 4.1 describes the simulation parameters, and 4.2 provides experimental data collected from the real sensors along with a comparison with the simulation results.

### 4.1 Simulation based Dataset

We considered five morning activities performed daily, which are: sleeping, getting out through a door, getting in through a door, relaxing on a couch and having breakfast. In each activity, we deployed three types of sensors – vibration sensor, force sensor, and RFID tags. Table 4 provides a labeled list of all the sensors used in the simulation experiments. Since our goal is to determine the effective types and numbers of sensors, we considered several possible sensor combinations. In this experiment, all sensors have uniform distribution with 0 mean and a variance of 1. RFID tags generate only 0 or 1 values (1 when a tag is detected).

**Table 4.** Sensors and RFID tags for each activity.

| Activity | Vibration Sensor | Force Sensor | RFID Tags |
|---|---|---|---|
| Sleeping | V01, V02, V03 | F01, F02 | |
| Getting out/in | | F03, F04 | RFID_Door01 |
| Relaxing | V04, V05, V06 | | |
| Eating | | F05 | RFID_Fork 01 & 02<br>RFID_Knife 01& 02<br>RFID_Spoon 01& 02 |

Parameters generated by an activity recognition algorithm are provided as inputs to the simulator. In our experiment, we applied neural network based activity recognition algorithm [1]. The algorithm is compared to the activity recognition algorithm in [10] and is shown to have better performance.

Table 5 shows simulation results for the *sleeping* activity. This activity is split into two actions: *Lying down* and *getting up*. Since *Lying down* is at most associated with 1 force sensor (F01) and 3 vibration sensors, we test 3 cases. Another force sensor, F02 detects a movement when a human subject gets up and puts his/her foot on the carpet that is attached to the sensor. Hence this sensor is used in every single case. The right-most column of Table 5 presents the simulation accuracy based on the activity recognition algorithm described previously. The algorithm [1] slices simulated events in a unit time (1 minute) so that the number of time slots the test case has been simulated can be determined. Then the algorithm can show how many time slots might be recognized among them. Therefore, their ratio, shown in the right-most column, indicates how many unit events are generated in a time window slot.

In these simulation experiments, we assume that the sensors for case 2 are deployed between the positions of sensors for case 3. Therefore sensors at the edges in case 3 might have greater distance between each other. We think it will reduce the probability of detecting "vibration on the bed", and eventually the number of the recognized time slots is slightly less than the one in case 2. Hence, we might see that case 2 indicates higher efficiency than case 3, even though it has less vibration sensors than case 3. Also, case 2 will be efficient in terms of cost of devices.

**Table 5.** The *sleeping* activity simulation cases.

| Case # | Number of Vibration Sensors | Number of Force Sensors | Number of Detections | Number of 1 min. Slots | Number of Activities Recognized | Accuracy |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 46 | 20 | 14 | 70.0% |
| 2 | 2 | 2 | 49 | 20 | 17 | 85.0% |
| 3 | 3 | 2 | 37 | 20 | 16 | 80.0% |

For *eating* activity results, we have four cases resulting from the combination of one force sensor and two RFID readers. Their simulation results are shown in Table 6. When only RFIDs are used to sense the activity, intuitively these cases cannot recognize the activity well because sensing some tableware does not always involve eating. In the worst case, for instance, a human may intends to eat and hence prepares utensils, but does not actually eat anything. A force sensor, F05 could help to detect the activity accurately, since it is triggered by change of force on the plate. For instance, when the user scoops mashed potato on the plate, the force sensor detects it and we could conclude that eating is happening with a spoon sensed by an RFID reader.

**Table 6.** The *eating* activity simulation cases.

| Case # | Number of RFID Readers | Number of RFID Tags | Number of Force Sensors | Number of Detection | Number of 1 min. Slots | Number of Activity Recognized | Accuracy |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 0 | 38 | 15 | 0 | 0.0% |
| 2 | 2 | 3 | 0 | 50 | 15 | 0 | 0.0% |
| 3 | 1 | 3 | 1 | 59 | 15 | 12 | 80.0% |
| 4 | 2 | 3 | 1 | 102 | 15 | 13 | 86.7% |

The test case with two RFID readers and one force sensor shows good accuracy, which is 86.7%. However, when we use only one RFID reader, the accuracy does not drop by much (drops to only 80.0%). Even though the first case has better performance, it costs almost twice as much as the case with a single RFID reader. Trading off little performance for significant cost reduction, the one RFID and one force sensor case could be most effective.

### 4.2 Verification by Real Sensor Deployment

In this section, we verify our two-phase approach by experiments with real sensors. The real sensor set is the same as that used for the simulation in section 4.1. We implemented and executed an activity recognition algorithm to compare the performance of dataset. The activity recognition system has 96.9 % accuracy performance which is better than that of an activity recognition system in [10]. The real sensor dataset is compared to the simulation dataset.

In Table 7 the performance of sleeping datasets is shown. The datasets represent 20 minutes of sleeping activity. Both simulation and real dataset show that the second case is more effective than other cases. However, simulation dataset shows better performance than the real sensor dataset.

**Table 7.** Comparison of *sleeping* activity. It presents accuracy between simulation and real sensor datasets for *sleeping* activity.

| Case # | Number of Vibration Sensor | Number of Force Sensor (Bed) | Number of Force Sensor (Floor) | Accuracy of Simulation Dataset | Accuracy of Real Sensors Dataset |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 70.0 % | 65.0 % |
| 2 | 2 | 1 | 1 | 85.0 % | 80.0 % |
| 3 | 3 | 1 | 1 | 80.0 % | 75.0 % |

Table 8 shows the *eating* activity datasets which are performed 15 minutes for each case. It compares the performance of simulation dataset with real sensor dataset. To filter out false *eating* activity, the activity recognition system does not consider the *eating* activity to be executed if there is only a single action in a specific time slot (e.g. one minute). For example, if there is only *serving* action or *cutting* in a time slot, it is unlikely that the person is actually eating the food. Both cases show that force sensor is crucial for detecting *eating* activity. Table 8 shows that the dataset of both cases exhibit better performance when the number of RFID readers increased. However, the effect of increasing the number of RFID readers is less than 7%.

**Table 8.** Comparison of *eating* activity datasets. It presents accuracy between simulation and real sensor datasets for *sleeping* activity.

| Case # | Number of RFID Readers | Number of RFID Tags | Number of Force Sensor (Plate) | Accuracy of Simulation Dataset | Accuracy of Real Sensors Dataset |
|---|---|---|---|---|---|
| 1 | 1 | 3 | 0 | 0.0 % | 0.0 % |
| 2 | 2 | 3 | 0 | 0.0 % | 0.0 % |
| 3 | 1 | 3 | 1 | 80.0 % | 93.3 % |
| 4 | 2 | 3 | 1 | 86.7 % | 100.0 % |

In Table 7 and Table 8, we observe that *eating* activity dataset shows better performance than *sleeping* activity. It means *sleeping* is more difficult to detect than *eating*. This is because of the inherent features of the *sleeping* activity. To illustrate, when people sleep well, there is very little action that can be detected by a sensor.

However, when people eat well, it usually translates into greater physical activity. Since activity is recognized depending on the performance of their actions, *eating* activity is easier to detect than *sleeping* activity.

## 5 Conclusion

Activity recognition research requires availability of datasets for testing and evaluation of recognition algorithms and for better understanding of human activities. However, it is often difficult to acquire or establish suitable datasets because they may simply not exist or if they have to be constructed, they require a lot of effort, time, and cost. In order to alleviate this problem, we proposed a two-phase activity dataset design approach, which is composed of three steps. To validate our approach, we implemented both a simulator and a real sensor environments for *sleeping* and *eating* activities. We compared the simulation datasets with real sensor datasets and showed similar results in terms of performance and effectiveness of the sensor set chosen according to our design approach.

A major advantage of the proposed approach is that researchers can acquire the most effective sensor configuration through well-controlled simulations instead of repeated real-world sensor deployment/assessment loops. This allows researchers to test novel activity recognition algorithms and activity models quickly, cost-effectively and accurately. Therefore, this approach can potentially accelerate the research and development of activity recognition systems.

## References

1. Helal, S., Kim E. and Hossain, S. “Scalable Approaches to Activity Recognition Research,” Pervasive 2010.
2. Kim,E. J., Helal, A. and Diane Cook “Human Activity Recognition and Pattern Discovery, IEEE Pervasive Computing, p 48, 2010.
3. W. Mann, A. Helal, "Smart Technology: A Bright Future for Independent Living", The Society for Certified Senior Advisors Journal, Volume 21, December 2003.
4. S. S. Intille, K. Larson, E. Munguia Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research," in Proceedings of PERVASIVE 2006, vol. LNCS 3968, K. P. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds. Berlin Heidelberg: Springer-Verlag, 2006, pp. 349-365
5. E. Munguia Tapia, S.S. Intille, L. Lopez, and K. Larson, "The design of a portable kit of wireless sensors for naturalistic data collection" in Proceedings of PERVASIVE 2006. Berlin Heidelberg: Springer-Verlag, 2006
6. B. Logan, J. Healey, Matthai Philipose, E. Munguia Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," in Proceedings of the International Conference on Ubiquitous Computing. vol. LNCS 4717 Berlin Heidelberg: Springer-Verlag, 2007, pp. 483–500.
7. Washington State University Activity dataset, http://ailab.eecs.wsu.edu/casas/datasets.html
8. A. Crandall and D. Cook. Coping with multiple residents in a smart environment. Journal of Ambient Intelligence and Smart Environments
9. Phidgets plug & play sensors, http://www.phidgets.com/

10. Kasteren, T. Noulas, A. Englebienne, G.and Krose, B. "Accurate Activity Recognition in a Home Setting," Proceedings of the Tenth International Conference on Ubiquitous Computing (Ubicomp 2008), Seoul, Korea, pp 1-9, 2008.
11. S. Hossain, A. Helal, H. Hagras, JW Lee and D. Cook, "Persim- Simulator for Human Activities in Pervasive Spaces," Submitted to the 12th International Conference on Ubiquitous Computing (Ubicomp) to be held Sept 26-29, 2010, Copenhagen, Denmark.
12. A. Elfaham, H. Hagras, A. Helal, S. Hossain, JW Lee and D. Cook, "A Fuzzy Based Verification Agent for the PerSim Human Activity Simulator in Ambient Intelligent Environments," Proceedings of the IEEE World Congress on Computational Intelligence, July 18-23, 2010, Barcelona, Spain.
13. A. Helal, A. Mendez-Vazquez, S. Hossain , "Specification and Synthesis of Sensory Datasets in Pervasive Spaces", In Proceedings of the IEEE Symposium on Computers and Communications (ISCC'09), Sousse, Tunisia, July 5-8, 2009.
14. J. N. Al-Karaki and G. A. Al-Mashaqbeh, "SENSORIA: A New Simulation Platform for Wireless Sensor Networks," Proceedings of the 2007 International Conference on Sensor Technologies and Applications, pages 424-429. IEEE Computer Society, October 2007.
15. S. Hossain, A. Helal, J.W. Lee, H. Hagras, A. Alfaham and D. Cook, "PerSim – A Simulator for Human Activities in Pervasive Spaces," University of Florida Technical Report number REP-2010-485, January 2010.
16. W. Jouve, J. Bruneau and C. Consel, "DiaSim: A parameterized simulator for pervasive computing applications," In PERCOM '09: Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications, March 2009, pp.1-3.
17. T. Antoine-Santoni, J. Santucci, E. De Gentili and B. Costa, "Modelling & simulation oriented components of wireless sensor network using DEVS formalism," In SpringSim '07: Proceedings of the 2007 spring simulation multiconference, March 2007, pp. 299-306.
18. Persim – A simulator for human-activities in Pervasive Spaces. Project web site: http://www.icta.ufl.edu/projects_persim.
19. SDDL – Sensory Dataset Description Language. Specifications and resources web site: www.icta.ufl.edu/persim/sddl/
20. M. Varshney and R. Bagrodia, "Detailed Models for Sensor Network Simulations and their Impact on Network Performance," Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, pages 70-77. ACM, October 2004.