An Empirical Approach to Modeling User-System Interaction Conflicts in Smart Homes

Fereshteh Jadidi Miandashti, Mohammad Izadi, Ali Asghar Nazari Shirehjini, and Shervin Shirmohammadi, *Fellow, IEEE*

***Abstract*—Conflict is one of the important factors affecting user satisfaction and trust in smart environments, yet conflict modeling in mixed initiative smart environments has not been sufficiently explored. Most of the existing literature on conflict in smart homes are centered on conflicts between users. Although research has shown that about 75% of conflicts are between users and system [1], only a few studies have considered user-system conflicts in smart homes. The aim of this article is to empirically propose both a definition and a run-time detection method for conflicts between users and smart home systems. Our empirical study is based on conflict sample scenarios collected from 163 users. Using clustering on these scenarios, we form an empirical definition of user-system conflict in smart homes. We also propose two functions that characterize each class of the collected scenarios, and we detect conflicts from this characterization. Our conflict detection model could help users achieve a more satisfactory experience in smart homes. Moreover, the model can offer benefits for system developers to design and deploy more reliable smart homes.**

***Index Terms*—Automated home, conflict repository, empirical study, human computer interaction (HCI), human machine interaction, mixed initiative smart homes, user-space conflict, user-system conflict, smart home conflicts.**

I. Introduction

AMBIENT Intelligence (AmI) is a domain in which devices, tools, and the environment are networked in a way that they can interact with each other and perform actions based on users' needs. "In an AmI world, massively distributed devices operate collectively while embedded in the environment using information and intelligence that is hidden in the interconnection network" [2]. The aim of this domain is to provide a proactive environment that is aware of the users' presence, as well as their personal, emotional, and locational characteristics, and is able to respond to users' needs properly [3].

A practical example of user-centric ambient intelligent environments is a smart home, which provides more comfort, performs day to day tasks, and increases safety [4]–[10]. In order to achieve this, the smart home needs to keep the appliances, services, applications, and users synchronous. The variety of affective factors in such environments on the one hand, and the dynamic nature of these factors on the other, can cause different complications. One of the most important of these complications is the notion of conflict [11].

Smart home users not only expect to be able to control their home easily, but also expect their home to adapt itself to their needs, actions, and preferences in time. Therefore, whenever the home fails to perform as expected, conflict happens, and it leads to users' dissatisfaction [12], [13].

Conflicts happen when users cannot perceive the intended results of their recent actions because of some confluence or interference of other actions that were triggered by the system in the same context. As Don Norman has observed [14], humans execute "actions" because they want to cause "effects" in their "world" and reach their "do-goals." Interaction conflicts undo, change, shift, or prevent these intended effects. Conflict is a subjective concept; it is a situation in which the user perceives the automation behavior as wrong or inappropriate. This is in contrast to errors, which appear in a system due to hardware or software malfunction, programming errors, or logical errors that were imposed at the design-time. Such errors are not the focus of this article.

Suppose that during the day the smart home system dims the lights in order to save energy despite the fact that the user prefers more light for reading. This is an example of conflict between user and system, and the type of conflict that we address in this article. As can be seen in the example, the existence of different contexts in a smart environment can lead to conflicts. Research shows that the occurrence of conflicts in such systems decreases users' trust [15], [16] and leads to dissatisfaction.

A general understanding of conflict is not necessarily enough for successful conflict detection because the meaning of conflict can be different for each application domain; therefore, it needs to be defined precisely according to that domain. In this article, we focus on detecting a conflict from users' point of view in a smart home context at run-time. Most of the research on the definition and detection of conflict in smart environments have focused on either conflicts between users [17]–[23], or conflicts between subsystems [18], [20], [22], [24]–[28], but few have attempted to define and model conflicts between the user and the system [11], [24], [29], [30], and even the latter research works have only studied specific domains such as lighting and heating, or media services [11], [24], [29]. Furthermore, many

Manuscript received August 20, 2019; revised January 31, 2020 and May 17, 2020; accepted July 18, 2020. Date of publication September 29, 2020; date of current version November 12, 2020. This article was recommended by Associate Editor L. L. Chen. *(Corresponding author: Ali Asghar Nazari Shirehjini.)*

Fereshteh Jadidi Miandashti and Mohammad Izadi are with the Sharif University of Technology, Tehran 11365-11155, Iran (e-mail: jadidif@ce.sharif.edu; lizadi@sharif.edu).

Ali Asghar Nazari Shirehjini and Shervin Shirmohammadi are with the University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: anazaris@uottawa.ca; shervin@ieee.org).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2020.3017784

2168-2291 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

of these studies were conducted in smart environments other than smart homes [25], [26], [30], [31], [32]. In an attempt to fill in the existing gap in this area, in this article, we make the following contributions.

1) We provide an empirical definition for user-system conflicts in smart homes.
2) We present a model to detect conflicts between the user and the system in a broad range of smart home subsystems.
3) We also offer to researchers in this field a repository [33] of 102 conflict scenarios which we have collected from users' point of view, and which can be used for smart home usability tests in the future.

Our study focuses on mixed-initiative smart home systems. "Mixed-initiative interaction has been proposed as an interaction strategy where members of a team can interleave their control of a mission based on their respective skill sets" [34]. In order to attain our goals, a qualitative research, guided by empirical user research principles, was conducted through semistructured interviews and focus groups. Using the concept mapping method [35]–[40], a repository of conflict scenarios was gathered from 163 people, and was then clustered by 11 experts. Feature extraction was later performed on the scenarios, and using linear discriminant analysis (LDA) [41] a model for conflict recognition was built. LDA is a popular data analysis technique for classification and feature selection. The output of a discriminant analysis method such as LDA is a set of discriminant functions, which are linear combinations of independent variables that create the best discriminate among different classes [42]. LDA has been applied successfully in many studies of human behavior recognition in smart environments [42]–[46].

The remainder of this article is organized as follows. Section II reviews related studies. Section III describes our methodology and research design. Section IV and Section V present the research results and discussion, and limitations, respectively. Finally, the conclusion and future works are summarized in Section VI.

II. Related Work

To this day there has been many research works on detection of conflicts in general, but only few have investigated this in the smart home domain [6]. The issue of conflict detection in smart environments is still considered a topic that needs more exploration [47]. In this section, a review of existing approaches for conflict detection in pervasive computing and Aml, particularly in the smart home domain, is presented. We can see that there is no universally agreed upon definition of conflict, and conflict seems to be application/situation specific and subjective.

A. Conflict and Its Categories

Webster's dictionary defines "conflict" as "competitive or opposing action of incompatibles (as of divergent ideas, interests, or persons) "or "mental struggle resulting from incompatible or opposing needs, drives, wishes, or external or internal demands". Based on Tuttlies's findings [48] "conflict occurs for an application or a user, if a context change leads to a state of the environment which is considered inadmissible by the application or user." According to [49] conflict is "the impossibility of an agent or group of agents reaching a goal that matters." Zoller defined conflict as a situation where processing a new request might lead to unwanted results because of a recently processed command with an opposite intent [29]. In another research [16], he pointed that conflicts occur when automation performs exactly the opposite of a user's action or performs an action that causes some parameter to be outside of the desired range for the user.

In smart environments, some actions might influence environment variables as well as change device status. Conflicts on environment variables are defined as situations when some services oppose other services in changing the value of these variables [24], [18]. Armac suggested that concurrency on the same entity and overlapping of incompatible requirements, features, or behaviors, are two common premises for conflict situations [24]. Resendes provided an extensive review about conflicts in smart environments and extended these definitions by proposing a rich taxonomy for conflicts [50], categorizing conflicts based on four dimensions.

1) Source.
2) Intervenient.
3) Time of detection.
4) Solvability.

According to this taxonomy, conflicts can be classified into two groups regarding their intervenient: user-user conflicts, where the intentions of a single user or multiple users contradict; or user-system conflicts, where system's actions are inconsistent with users' actions or intentions.

Other researchers also proposed another type of conflict that occurs between services or applications which are considered a type of system-system conflict [26], [51], [47]. In system-system conflicts, actions of different subsystems contradict. One type of system-system conflict is application-application conflict, which is concerned with conflicts between user applications.

Based on the results of analyzing 40 devices and behaviors related to them in smart homes, Armac *et al.* [24] classified conflicts in smart homes into four groups:

1) Conflict between services that leads to user-user conflict, while using equipment that can be used by only one person at a time.
2) Conflict between services that use common resources.
3) Conflicts caused by changes in the environment variables.
4) Conflicts that occur when users with different priorities using the same resources.

In summary, a conflict situation regarding its intervenient can be either user-user, user-system or system-system conflict.

B. Conflict Detection

Here we present the existing research in conflict detection for the following types of conflicts: user-user, system-system (app-app or service-service), and user-system.

1) User-User Conflict Detection: User-user conflicts have been widely investigated in the domain of smart environments. Hanssens *et al*. [31] studied user-user conflict detection in intelligent business rooms. In their research, a series of behaviors were activated based on the order of user actions. Behaviors were layered upon each other based on their level of generality. Contradicting behaviors in the same layer were detected as conflicts.

Furthermore, Shin *et al.* [21]–[23] proposed methods for user-user conflict detection in media services at smart homes. They developed a context manager [52], which detects service conflicts. In fact, they investigated conflicts that occur between services, which leads to conflicts between users. Conflicts among services were detected by utilizing user preferences and service properties in their methods.

In a study by Roy *et al.* [19], a framework for user-user conflicts in smart homes was proposed. This framework is based on game theory and considers the users as players and finds each user's location. This method predicts conflicts before occurrence.

Tuttlies *et al.* [48] presented a model for potential user-user conflict avoidance including a database of conflict situations presented in a specification language. This database is constantly updated by different applications and users. In their model, the effect of each action on the environment is assessed before the action takes place, and is compared to database contents in order to decide if it leads to a conflict situation in real time.

Kung and Lin [25] designed a context-aware embedded multimedia presentation system. This system includes three main mechanisms: A mechanism for context detection, a mechanism for conflict detection based on case-based reasoning rules, and a mechanism for conflict resolution. This approach focused on user-user conflicts and application-application conflicts, which in different circumstances can lead to one another.

Braga Silva *et al.* [27], [28] introduced an approach to dynamically detect and solve conflicts for collective, ubiquitous and context-aware applications. Their main aim was to improve the tradeoff between user satisfaction and resources usage. Their approach was designed for the detection of conflicts between users. Their testbed was a context-aware tourist application.

2) System-System Conflict Detection: Munir and Stankovic [53] proposed an approach for system-system conflict detection in sensors and actuators in home settings. Their approach, detects conflicts at runtime by considering the dependency between applications, sensors, and actuators.

In 2019, Farooq *et al.* [47] provided a formal approach for service conflict detection in IoT systems. They defined safety properties of an IoT system and proposed a technique for detecting events that violate the safety properties. They implemented a simulated smart home, IoTC^2, as their test bed and conducted some experiments to test safety properties violation of controllers and actuators' behavior.

Shin *et al.* in [20] proposed a method utilizing properties of the services and users' preferences to detect conflicts among services. They applied Bayesian theory to conflict history of users and applications and adjust weights of the context.

Park *et al.* [18] proposed a system-system and also user-user conflict management scheme for group-aware ubiquitous computing environments. The proposed scheme can detect conflicts based on changes in environment variables. Each action semantic was specified with a rule definition language and presented to a first-order predicate logic inference that was utilized to capture conflict.

3) User-System Conflict Detection: Some studies have proposed methods for detecting user-system conflicts in a variety of domains. Pizziol *et al.* [30] proposed a generic formal approach based on Petri nets in order to model user-system conflicts in the aviation domain. This model was used to detect potential pilot-automation conflicts. Pizziol used this method because possible procedures and conditions are finite and prespecified in the aviation domain.

Armac *et al.* [24] proposed an approach for the detection of user-user, service-service and also user-system conflicts in smart homes. This method discovers conflicts at runtime and before occurrence. They analyzed more than 40 devices and behaviors related to them in smart homes. Based on the results of this analysis, they classified conflicts in smart homes. Each type of conflict was specified in a formal rule-based language and was detected with the rule-based model. However, only the conflict between services which leads to user-user conflict type was addressed and results for the other conflict types have not been presented.

Sun *et al.* [11] presented a conflict detection algorithm based on a formal rule model. The rules were based on users, triggers, environment entities, and actuators. Their model tailored for smart buildings with multiple users and detects user-system and also system-system conflicts. This approach is effective but with low efficiency sometimes, as stated by the authors themselves in their conclusion section [11].

Zoller and Nazari Shirehjini [29] attempted to define and detect conflicts between the user and the system in smart homes. Their approach checked for contradicting user intentions and system action effects on environment variables in a predefined time duration.

Several conflict detection approaches in smart homes are given in Table I. As we can see, most of the existing literature on conflict in smart homes is centered on user-user or service-service conflicts. Whereas related research indicates that user-system conflicts decrease trust and user satisfaction [12], [13], [15], [16], only a few studies on user-system conflict have been reported. Therefore, we based our research on modeling and run-time detection of conflicts between users and smart home systems.

III. Method

Our research methodology consisted of two main phases: user-system conflict definition in smart homes, and designing a user-system conflict detection model for smart homes. The first phase included a two-step user research. In the first step, we collected scenarios of conflicts from nonexpert users, and in the second step, experts validated, rated, and sorted the previously collected conflict scenarios. The second phase consisted of data analysis in order to design a model for conflict detection.

A. Experimental Design and Procedure

Since one of our research goals was to define conflict, a qualitative approach was used. Generally, qualitative research is

TABLE I
LITERATURE STUDY AND COMPARISONS OF CONFLICT MANAGEMENT APPROACHES IN SMART HOMES

Author	Conflict Type	Conflict Formal Specification	Conflict Detection	Approach
Park *et al.*[18]	user-user (app-app)	X	X	Detection based on analyzing changes in environment variables. Resolution based on users' preferences weights
Roy et al.[19]	user-user		X	Resolution based on stochastic game theory
Shin *et al.*[23][21]	user-user		X	Resolution based on users' priorities and consensus
Shin *et al.*[22]	user-user		X	A Mixed-initiative conflict resolution approach (user mediated resolution approach and automatic approach)
Thyagaraju *et al.* [17]	user-user		X	Application developers and end-users specify conflicts situations. Resolution based on users' priorities
Bikakis *et al.*[26]	user-user, system-system	X	X	Handling inconsistency in distributed context knowledge
Camacho *et al.* [66]	user-user		X	Knowledge-based analysis
Shin *et al.*[20]	service-service (user-user, app-app)		X	Utilizing conflict history of users as well as user contexts and service profiles
Retkowitz *et al.* [51]	service-service (user-user, app-app)		X	Detection with analyzing dependency between services. Resolution based on users' priority groups.
Munir *et al.*	system-system	X	X	Detecting conflicts at runtime by considering the dependency between applications, sensors, and actuators.
Sun *et al.* [11]	***user-system,*** service-service	X	X	Rule based conflict detection approach
Farooq *et al.* [47]	service-service	X	X	Formal method approach, to ensure safety with respect to conflicts.
Armac *et al.*[24]	***user-system,*** user-user, system-system	X	X	Rule based conflict detection approach, Interest/intention conflict Resolution
Zoller [29]	***user-system***		X	Checking user intention with system action effects on environment variables

used for illuminating the meaning of a concept [54]. The other goal of our research was modeling conflict detection; therefore, we also conducted a quantitative research. The phases and steps of our research methodology are shown in Fig. 1.

As we mentioned earlier, the first phase of the experiment consisted of two main steps: In the first step, we conducted interviews and focus groups in order to collect scenarios. In the second step, we gave the scenarios to 11 experts for rating and sorting them into groups. The experts had solid knowledge of Aml or human computer interaction (HCI). They were members of the HCI or the AmI laboratories of various universities, and we recruited them via email in the first phase. After these two steps, we performed several analyses based on the concept mapping method in order to cluster the conflict scenarios.

Following, we describe the experiment design in detail. First, we recruited some users. While demographic and users' technology affinity are important factors in the experiment results, they are not important for recruitment because anybody could be a smart home user. After greeting the interviewee and giving a brief description of our research method and the experimental process, they were asked to sign a consent form and fill out a questionnaire recording their demographic background. Then, we tried to develop an appropriate mental model of a smart home for the user (or the group of users), in about 10 min: first, explanation about smart home and its properties, features and functionalities were provided for the users. Then two short videos (3 and 4 min) about smart homes and their facilities were played for the users. We previously verified the effects

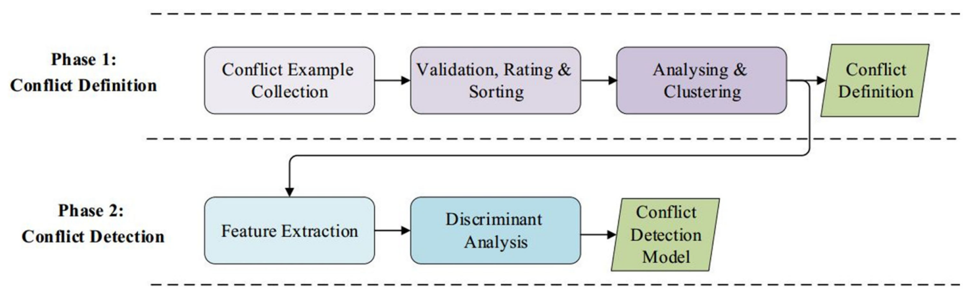


Fig. 1. Research method.

of the above method for mental model development in another experiment. We conducted the test with 12 nonexperienced subjects. A before–after open card-sorting method [55] was used in order to analyze to which extent their mental models had changed after watching the videos and listening to the experimenter's explanation. The results of this test were satisfactory.

Next, the following informal definition of conflict in smart environments was presented to the participant: "When a user's needs or preferences are not satisfied by the smart home, conflict occurs."

The next step was conflict scenario collection. The users were asked to present at least two user-system conflict scenario examples that may occur in a smart home. If the user could give the scenario examples within 10 min, then we used the user's scenario examples. Otherwise, the following conflict examples were presented to them.

1) The air conditioner is on, and the user, to enjoy fresh air, opens the window. However, the system closes the window immediately in order to save energy.
2) You have programmed the system to do many energy consuming tasks, such as washing the dishes, washing the clothes, recording your favorite movie, cooking, etc., such that they are consuming altogether more energy than the user's specific threshold. Because the energy usage is beyond the threshold, the system would not wash the clothes whereas you needed them.
3) The system has correctly learned what the user's favorite TV program is. It therefore automatically turns on the TV at the start of the user's favorite TV program, without the user having asked for it, and while the user is listening to music through the same sound system.
4) A between user conflicts arises when one user wants to have the window open while another user wants it closed.
5) The amount of needed light for two roommates at night, when the first user wants to rest and the second user wants to study, is contradictory.

After providing these examples, the experimenter pointed out that the last two are conflict between users, whereas we intended to collect user-system conflicts. Again, the user was asked to present a few conflict scenarios.

The process was improved in an iterative fashion and biases were reduced as much as possible.

B. *Participants*

Since smart home is a user-centric product, we intended to define conflict based on its users' opinions and experiences. Since smart home technologies are not widely adopted, and the few real users of smart homes are often not accessible, we decided to interview ordinary people who were potential users of smart homes in order to reach an adequate sample size.

In the first step of our user research, 163 nonexpert participants aged between 21 and 68 years old were recruited, among which were physicians, teachers, authors ($N = 43$, 12 male, 31 female, mean age 41.7 SD = 9.9) and 120 computer students aged between 20 and 25. Nonexpert users were interviewed individually or in focus groups after development of an appropriate mental model using videos and explanations by the experimenter. The interviews took place in the participant's place or in the laboratory and did not need any prearranged setting. The interviews were recorded with the interviewee's permission for later review purposes. The study lasted for seven months.

C. *Data Collection*

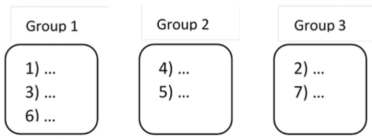
After recording demographic background of participants and developing the mental model for them, they were asked to present at least two conflict scenario examples.

The collected conflict scenarios were illustrated by a graphic designer and then approved by the users, to ensure that the experimenter correctly understood what the participants meant.

When all interviews finished, data aggregation and cleaning were done and duplicates were omitted. Some of the scenarios presented by users did not indicate a conflict situation; these scenarios were eliminated after the approval of three of the five experts. Also, conflict scenarios other than user-system conflicts were excluded. Data gathering resulted in 153 statements that were edited and synthesized into a final set of 102 scenarios.

D. *Data Analysis*

The final set of scenarios was then given to 11 HCI or AmI experts, individually. It should be noted that in exploratory user studies, a sample size of 2-3 is considered adequate [56], [57]. First, the concept of smart home and its properties, features and functionalities were briefly introduced to the experts. Then, we used word count as a measure for verifying the effects of our



	1	2	3	4	5	6	7
1	1	0	1	0	0	1	0
2	0	1	0	0	0	0	1
3	1	0	1	0	0	1	0
4	0	0	0	1	1	0	0
5	0	0	0	1	1	0	0
6	1	0	1	0	0	1	0
7	0	1	0	0	0	0	1

Fig. 2. Seven scenarios grouped in three piles and corresponding similarity matrix example.

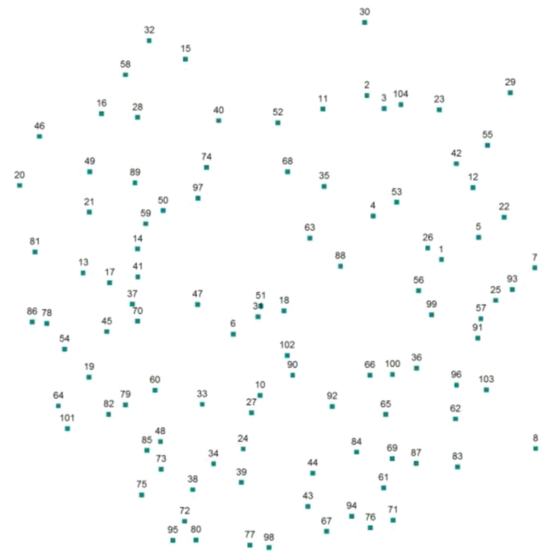


Fig. 3. Point map derived from MDS of the example scenarios.

studies, a sample size of 2-3 is considered adequate [56], [57]. First, the concept of smart home and its properties, features, and functionalities were briefly introduced to the experts. Then, we used word count as a measure for verifying the effects of our explanations [58], [59], [60].

Next, the experts were presented with the previously collected conflict scenarios and asked which scenarios they consider as real conflict situations, and to rate the intensity of each on a five-point Likert scale (point 1 is the weakest form of conflict, and point 5 is the strongest form of conflict). After rating, the experts had to group the conflict scenarios in arbitrary disjoint categories. In concept mapping, this method is called sorting. The number of groups formed by the experts ranged from 3 to 7. An interesting observation, which shows the subjectiveness of conflict types, is that each expert had his/her own different point of view for grouping the scenarios. For example, expert 1 grouped the cases into three categories: individual, social, and financial conflicts, but expert 2 grouped the same cases into three dimensions: electronic device based events, electrical device-based events, and mechanical device-based events.

Data analysis of the first phase was based on the concept mapping method [61]. After the experts completed sorting the conflict scenarios, their results were aggregated in the following steps.

1) First, a symmetric similarity matrix was produced for each expert's sorting result. An example is presented in Fig. 2. Suppose that we had seven scenarios and one of the experts grouped them into three piles. A 7 $\times$ 7-similarity matrix is created for the expert. Cell values (0 or 1) indicate whether a pair of scenarios was grouped into the same pile or not.
2) Second, the summation of all the individual matrices was obtained. This new summation matrix indicates how many experts have put each pair of scenarios together in one category, regardless of what the categories' meant to each expert. For example, if $m_{i,j}$ in the aggregated matrix is equal to 5, it means that five experts believe scenario *i* and scenario *j* are in the same group. The summation matrix provides information about the participants' classification of the conflict scenarios and therefore indicates the relational structure of the conceptual domain.
3) Third, a point map is developed by a multidimensional scaling algorithm (MDS) [62], [63] taking the group similarity matrix. In this point map, as shown in Fig. 3, each point indicates a conflict scenario, and the closer the points are to each other on the map, the more people have grouped the corresponding scenarios together. In other words, points that are more similar are closer to each other on the map. Distances between points are important, not the position of each point on the map.

Next, a hierarchical cluster analysis was performed on the point map. Ward's hierarchical cluster analysis has been suggested by Trochim and McLinden [61] as it can generate different configurations of clusters, from which the experts can later choose the most sensible classification. The result was a cluster map called "the conceptual domain of scenarios" that partitioned conflict scenarios into groups where the members of each group were most similar to each other. Since there is no distinct mathematical criterion to find the optimal number of clusters, it requires judgment [64]. Therefore, the experts gathered in a session for group agreement and adjusting the number of clusters. The decision was made among the output of hierarchical clustering, based on the similarity of the scenarios as well as theoretical methods of interpretation and validation of cluster analysis consistency, such as Elbow and Euclidean. A final three-cluster solution was chosen as it both fitted the data and yielded interpretable results.

After proposing a definition for conflict, we designed a user-system conflict detection model in the second phase of the article. In this phase, we represented each scenario with a feature vector consisting of a number of features.

A group of 5 out of the 11 experts volunteered to work on performing feature extraction on the scenario examples obtained

TABLE II
FEATURES EXTRACTED FROM CONFLICT SCENARIOS

Scenario Features			
1.	usr_presence	14.	device1
2.	usr_past_action	15.	device2
3.	usr_Activity	16.	conflict_scenario_trigger
4.	usr_meanwhile_action	17.	conflict_reason
5.	usr_meanwhile_action_eff	18.	usr_complain
6.	usr_preference	19.	intensity
7.	usr_intention	20.	time_cost
8.	user_def_rule	21.	recoverable
9.	interaction	22.	usr_global_satisfaction
10.	sys_action	23.	special_case
11.	sys_action_eff	24.	time_condition
12.	sys_reason	25.	time_freq
13.	location		

in the previous phase. First, feature engineering was performed by consensus experts in multiple sessions. They extracted features from raw data considering key variables in each scenario. A total of 25 features were extracted, by the five experts, as given in Table II. These features characterize each scenario based on factors, such as user's presence, user's action, user's action effect, user's preference, system's action, and system's action effect. In fact, the features are predictor variables for determining which unknown scenario belongs to which cluster. Then, each expert individually extracted feature values for 30 scenarios (10 scenarios were common for all of the experts). Finally, we measured inter-rater reliability by calculating Cohen's kappa [65] and gained an acceptable level of agreement ($k = 0.89$).

Next, we used canonical discriminant analysis in order to find a linear combination of the extracted features, which characterizes each class of the collected scenarios. Scenario features that best distinguished between the three scenario groups were selected in order to obtain a discriminant function model. Before finding discriminant function coefficients, Wilk's lambda was used for multivariate statistical test of the differences between the scenario groups.

As given in Table III, out of the 25 features that were assigned to discriminant analysis using statistical package for the social sciences (SPSS) statistical software, six differed significantly in their cluster average values: usr_meanwhile_action_effect ($p = 0.004$), usr_preference ($p = 0.000$), interaction ($p = 0.004$), intensity ($p = 0.000$), time_cost ($p = 0.005$), and recoverable ($p = 0.009$).

In order to determine the weight of each feature in discriminant functions, we need to standardize them. Table IV gives the standardized features which are used to construct two canonical discriminant functions. The functions are the conflict detection model.

IV. RESULTS

In the first phase, as mentioned in Section III-D, the experts, based on the similarity of scenarios (as shown in Fig. 3, the point map), finally chose three clusters among the output of

TABLE III
SIGNIFICANCES OF FEATURES BETWEEN THE CLUSTERS

Feature	Wilks' Lambda	F (df1,df2)	Sig.
usr_presence	.971	1.448	.240
usr_past_action	.969	1.533	.221
usr_Activity	.953	2.415	.095
usr_meanwhile_action	.992	.410	.665
usr_meanwhile_action_eff	.892	5.893	.004
usr_preference	.825	10.276	.000
usr_intention	.998	.103	.902
user_def_rule	.976	1.169	.315
interaction	.891	5.945	.004
sys_action	.984	.771	.465
sys_action_eff	.943	2.951	.057
sys_reason	.903	5.200	.007
location	.962	1.935	.150
device1	.982	.888	.415
device2	.991	.446	.642
conflict_scenario_trigger	.966	1.705	.187
conflict_reason	.991	.443	.643
usr_complain	.954	2.361	.100
intensity	.781	13.630	.000
time_cost	.897	5.539	.005
recoverable	.908	4.940	.009
usr_global_satisfaction	.991	.458	.634
special_case	.964	1.799	.171
time_condition(event)	.952	2.472	.090
time_freq	.967	1.637	.200

TABLE IV
STANDARDIZED CANONICAL DISCRIMINANT FUNCTION COEFFICIENTS

Feature	Function 1 Coefficient	Function 2 Coefficient
usr_presence	.146	.498
usr_past_action	-.103	-.117
usr_Activity	.146	.513
usr_meanwhile_action	.063	-.152
usr_meanwhile_action_eff	-.325	-.164
usr_preference	-.347	.547
usr_intention	.237	-.346
user_def_rule	-.285	.014
interaction	.269	-.444
sys_action	-.011	-.365
sys_action_eff	.023	-.046
sys_reason	-.420	.237
location	-.215	.148
device1	-.105	.183
device2	-.088	.229
conflict_scenario_trigger	.309	.042
conflict_reason	-.048	-.085
usr_complain	.085	-.039
intensity	.544	.594
time_cost_	.302	-.156
recoverable	.178	-.065
usr_global_satisfaction	-.157	-.035
special_case	.094	-.032
time_condition	-.295	-.196
time_freq	.017	-.407

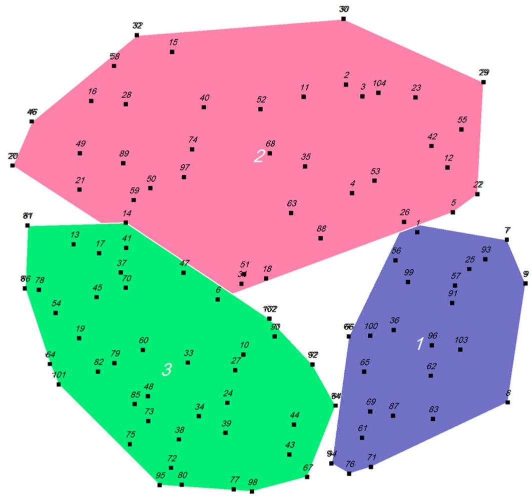


Fig. 4. Final clustering of the scenarios (an empirical definition of conflict in smart home).

TABLE V
FUNCTION AT CLUSTER CENTROIDS

Cluster	Function 1	Function 2
1	1.153	-.625
2	-1.496	-.453
3	.125	.817

hierarchical clustering. The result of clustering, illustrated in Fig. 4, formed an empirical definition of conflict in smart homes.

In the second phase, a canonical discriminant analysis was used on the data on which feature extraction was done, to build a prediction model. The model consists of two functions. Table IV gives the standardized canonical discriminant coefficients for each of the functions. These coefficients indicate the contribution of each feature to distinguishing between the three clusters. The following two functions were obtained using the coefficients.

$$\begin{aligned} \text{score1} &= 0.146{*}\text{user\_presence} \\ &\quad - 0.103{*}\text{user\_past\_action} + \ldots \qquad (1) \\ \text{score2} &= 0.498{*}\text{user\_presence} \\ &\quad - 0.117{*}\text{user\_past\_action} + \ . \qquad (2) \end{aligned}$$

With the two final discriminant functions, new scenarios with different values for each feature can be assigned to the equations. The difference between output scores and cluster centroids determine the cluster each scenarios belongs to. In other words, if the calculated scores for a scenario are closer to one of the clusters centroids, it means that the scenario is placed in that cluster. Functions at cluster centroids can be given in Table V.

These two functions are the conflict detector. Each situation in smart-home could be the input of the detector and the output is the label of: "conflict" or "we do not know if it is a conflict or not." The latter label is about scenarios that do not belong to our conflict clusters, so they may be some unknown sort of conflict or are not conflict at all.

TABLE VI
CLASSIFICATION RESULTS

		Predicted Group Membership			Total
	Cluster	1	2	3	
n	1	**23**	1	8	32
	2	2	**20**	6	28
	3	6	3	**31**	40
%	1	**71.9**	3.1	25.0	100.0
	2	7.1	**71.4**	21.4	100.0
	3	15.0	7.5	**77.5**	100.0

We used cross-validation to estimate the accuracy of the performance of our predictive model (see Table VI). In Table VI, of the 32 actual members of class1, the system predicted 23 scenarios correctly, and of the 28 members of the second class, it predicted that 20 belonged to the second class. All correct predictions are located in the diagonal of the table (highlighted in bold). It is easy to visually inspect Table VI for the percentage of prediction accuracy, in the last three rows. In summary, 74.0% of original grouped cases with cross-validation correctly classified.

V. Discussion

In this section, first we compare our approach with existing related works to show how some of our collected scenarios, subjectively perceived by users as being a conflict, do not fit with the definitions by those works. This enforces the statement in Section II that a universally agreed upon definition of conflict does not exist. Then, we describe how our results can be used by designers of smart home systems.

Armac *et al.* [24] addressed only the service-service conflict. Therefore, we could say that none of our user-provided cases satisfy their model because they are user-system conflicts.

Sun *et al.* [11] presented a conflict detection algorithm based on a formal rule model, addressing challenges including conflicting services customized by the users. Each rule requires some conditions to trigger rule execution. They proposed 11 rules for conflict detection, none of which contains unanticipated user's actions. If the user previously gave some conditions to the system, there are rules for detecting a conflict situation. But, if the user does any action whose conditions were not declared to the system before, the system would not check it for conflict. Therefore, all conflict cases like case 1 are not detected by Sun's rules.

Case 1: The air conditioner is on, but the user opens the window to enjoy the fresh air. This conflict could happen because the user forgot the window is open, or maybe the user did it on purpose. Irrespective of the reason, the system should detect this conflict and send it to the conflict resolution module of the system, which can either close the window immediately in order to save energy, or inform the user that the air conditioner is running.

Zoller and Nazari Shirehjini [29] attempted to define and detect conflicts between the user and the system in smart homes.

Their approach checked for contradicting user intentions and system action effects on environment variables (such as temperature, light, humidity, etc.). Therefore, all of the cases which are not about environment variables are not detected by their model, such as case 2.

Case 2: The system has correctly learned what the user's favorite TV program is. It therefore wants to automatically turn ON the TV when the program starts, without the user having asked for it. But, the sound system is tied up because the user is listening to music through the same sound system. This conflict could happen because the user completely forgot about the favorite program, or because the user is more interested in the music now. Irrespective of the reason, the system should detect this conflict and send it to the conflict resolution module of the system, which can either override the sound system and play the TV program, or inform the user that the TV program is about to start.

Besides, system designers could adopt our approach for a run-time detection of conflicts between users and smart home systems.

Despite the above differences, our repository could help designers of smart homes understand what rules (system behaviors) were perceived by users as conflict. This allows the designer to design appropriate conflict detection and conflict resolution modules for the system at hand. For example, the user might open the window to let in fresh air, but the user has forgotten that his pet bird is roaming free in the room and might escape from the window! The system knows the latter fact and closes the window immediately. A designer might come across a scenario similar to this, but in his/her own system. For example, the designer could face a scenario of "the user might leave the door open in anticipation of a guest's arrival, but there are puppies roaming freely in the house, so the system closes the door." In this case, the designer can use our repository to understand that his/her scenario, which is very similar to the said scenario in our repository, is indeed considered a conflict by smart home users; hence the designer can investigate it subjectively or with some well-known techniques, and design appropriate conflict detection and resolution rules for it in order to increase user satisfaction. In other words, each scenario in our repository could be a conflict situation, and by crosschecking the designer's system against our repository, the designer could improve the quality of design. Of course, our repository does not contain all possible conflict scenarios in a smart home, but it gives a fairly good starting point. More scenarios can increase the designer's chance to avoid more conflict situations.

In addition to using our repository, a designer can use our methodology itself to collect conflict scenarios from the users of the designer's system, incorporating those users' culture, common preferences, and perspectives.

We explained our repository to five senior designers and developers with more than ten years of experience in university and industry, and asked them: "What is the application of the repository for you and how will you use it?". The following is a summary of their answers.

They expressed that first, they would investigate the repository and extract the required knowledge by one of these methods.

1) Text mining.
2) Natural Language Processing techniques.
3) Defining some rules (and specify the rules by a logic).
4) Training a Neural Network.
5) Preparing a state-machine with events, actions, and then signatures.
6) Subjective investigation of each scenario.

Then, they would construct a conflict detector according to their extraction method.

VI. Limitations of the Study

This research proposes an empirical definition for user-system conflict and an approach for conflict detection in smart home. Even though we tried our best to design a thorough and accurate research method, some limitations still exist. First, since smart environments are still not widely adopted, the participants of this experiment were not real smart home users. Even though the accuracy of mental model development was checked by conventional methods, the results could potentially be of more quality with real smart home users. Furthermore, most of the participants were recruited among university students and were aged 20 to 25, therefore our sample lacked diversity. Moreover, the large number of conflict scenarios might have led to the reduction of the experts' precision in the grouping process.

VII. Conclusion

In this article, we proposed an empirical definition model for user-system conflicts in smart homes and presented an approach for conflict detection at run-time. Both of these can be useful in designing smart home middleware that improve user satisfaction. First, a database of conflict scenarios was collected from users after adequate mental model development. The scenarios were later grouped by 11 experts based on the concept mapping method. The resulted clusters constituted our empirical model. Second, we proposed a method to detect conflict in smart homes through contextual features generalized from the case studies. Feature extraction was performed on the example conflict scenarios by five of the experts, describing each scenario in vector form. Finally, conducting discriminant analysis on these vectors resulted in a conflict detection model with 74% precision.

Future work includes investigating user-system conflicts in multiuser smart homes. Furthermore, using subjects who are experienced with real smart home environments can improve the accuracy of our results. Moreover, the empirically derived model of conflict could be specified in formal logic and therefore conflict detection can be implemented by constraint programming. Additionally, with the repository of conflict scenarios, we could design an empirical acceptance score for smart homes. As an important future work we propose to describe our data and results using semantic web technologies. A smart home conflict ontology and semantic descriptions of the collected scenarios could support developers and smart home systems designers to easily access the insights we gained in this research at design time.

REFERENCES

[1] A. Hoseini, "*A New Approach in Interaction Conflict Cognition*," B.S. Thesis, Dept. Comput. Eng., Sharif Univ. Technol., Tehran, Iran, 2015.

[2] E. Aarts, and B. De Ruyter, "New research perspectives on ambient intelligence," *J. Ambient Intell. Smart Environ.*, vol. 1, no. 1, pp. 5–14, 2009, doi: 10.3233/AIS-2009-0001.

[3] M. Gams, I. Y. H. Gu, A. Härmä, A. Muñoz, and V. Tam, "Artificial intelligence and ambient intelligence," *J. Ambient Intell. Smart Environ.*, vol. 11, no. 1, pp. 71–86, 2019, doi: 10.3233/AIS-180508.

[4] M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homes—Present state and future challenges," *Comput. Methods Programs Biomed.*, vol. 91, no. 1, pp. 55–81, 2008, doi: 10.1016/j.cmpb.2008.02.001.

[5] M. Collotta and G. Pau, "Bluetooth for Internet of Things: A fuzzy approach to improve power management in smart homes," *Comput. Elect. Eng.*, vol. 44, pp. 137–152, 2015, doi: 10.1016/j.compeleceng.2015.01.005.

[6] P. Carreira, S. Resendes, and A. C. Santos, "Towards automatic conflict detection in home and building automation systems," *Pervasive Mobile Comput.*, vol. 12, pp. 37–57, 2014, doi: 10.1016/j.pmcj.2013.06.001.

[7] C. Y. Chen, J. H. Fu, T. Sung, P. Wang, E. Jou, and M. Feng, "Complex event processing for the Internet of Things and its applications," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2014, pp. 1144–1149.

[8] M. Bhole, K. Phull, A. Jose, and V. Lakkundi, "Delivering analytics services for smart homes," in *Proc. IEEE Conf. Wireless Sensors*, 2015, pp. 28–33, doi: 10.1109/ICWISE.2015.7380349.

[9] D. Bonino and F. Corno, "What would you ask to your home if it were intelligent? Exploring user expectations about next-generation homes," *J. Ambient Intell. Smart Environ.*, vol. 3, no. 2, pp. 111–126, 2011, doi: 10.3233/AIS-2011-0099.

[10] P. Kasnesis, "Collective domotic intelligence through dynamic injection of semantic rules," in *Proc. IEEE Int. Conf. Commun .*, 2015, pp. 592–597, doi: 10.1109/ICC.2015.7248386.

[11] Y. Sun, X. Wang, H. Luo, and X. Li, "Conflict detection scheme based on formal rule model for smart building systems," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 2, pp. 215–227, Apr. 2015, doi: 10.1109/THMS.2014.2364613.

[12] A. G. Mirnig, S. Trösterer, A. Meschtscherjakov, M. Gärtner, and M. Tscheligi, "Trust in automated vehicles," *I-Com J. Interact. Media*, vol. 17, no. 1, pp. 79–90, 2018, doi: 10.1515/icom-2017-0031.

[13] M. Ziefle, C. Röcker, and A. Holzinger, "Medical technology in smart homes: Exploring the user's perspective on privacy, intimacy and trust," in *Proc. 35th IEEE Annu. Comput. Softw. Appl. Conf. Workshops*, 2011, pp. 410–415, doi: 10.1109/COMPSACW.2011.75.

[14] D. A. Norman, *The Design of Everyday Things*, vol. 1. London, U.K.: Basic Books, 2013.

[15] A. Semsar and A. A. N. Shirehjini, "Multimedia-supported virtual experiment for online user- system trust studies," *Multimed. Syst.*, vol. 23, no. 5, pp. 583–597, 2017, doi: 10.1007/s00530-016-0519-4.

[16] A. Semsar and A. A. N. Shirehjini, "Analyzing the effect of interaction conflicts on trust in ambient intelligence home environments," Master Thesis, School Comput., Sharif Univ. Technol., Tehran, Iran, 2016.

[17] S. K. Thyagaraju, G. S. Joshi, S. M. Kulkarni, U. P. Narasimha Murthy and A. R. Yardi, "Conflict resolution in multiuser context-aware environments," in *Proc. IEEE Int. Conf. Comput. Intell. Model. Control Automat.*, 2008, pp. 332–338, doi: 10.1109/CIMCA.2008.87.

[18] I. Park, D. Lee, and S. J. Hyun, "A dynamic context-conflict management scheme for group-aware ubiquitous computing environments," in *Proc. 29th Annu. Int. Comput. Softw. Appl. Conf.*, 2005, pp. 42–47.

[19] N. Roy, A. Roy, and S. K. Das, "Context-aware resource management in multi-inhabitant smart homes: A nash h-learning based approach," in *Proc. 4th Annual IEEE Int. Conf. Pervasive Comput. Commun., PerCom* 2006, pp. 148–158, doi: 10.1109/PERCOM.2006.18.

[20] C. Shin, Y. Oh, and W. Woo, "History-based conflict management for multi-users and multi-services," in *Proc. Context 2005 Workshop Context Model. Decis. Support*, 2005, pp. 126–142.

[21] C. Shin, H. Yoon, and W. Woo, "User-centric conflict management for media services using personal companions," *ETRI J.*, vol. 29, no. 3, pp. 311–321, 2007, doi: 10.4218/etrij.07.0506.0023.

[22] C. Shin, A. K. Dey, W. Woo, and S. Korea, "Mixed-initiative conflict resolution for context-aware applications," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, 2008, pp. 262–271.

[23] C. Shin, A. K. Dey, and W. Woo, "Toward combining automatic resolution with social mediation for resolving multiuser conflicts," *Cybern. Syst.*, vol. 41, no. 2, pp. 146–166, 2010, doi: 10.1080/01969720903584282.

[24] I. Armac, M. Kirchhof, and L. Manolescu, "Modeling and analysis of functionality in eHome systems: Dynamic rule-based conflict detection," in *Proc. Int. Symp. Workshop Eng. Comput. Based Syst.*, 2006, no. Ecbs, pp. 219–228, doi: 10.1109/ECBS.2006.48.

[25] H. Kung and C. Lin, "Application-layer context-aware services for pervasive computing environments," in *Proc. IEEE 1st Int. Conf. Innov. Comput., Inf. Control*, 2006, vol. 3, pp. 229–232, doi: 10.1109/ICICIC.2006.423.

[26] A. Bikakis, G. Antoniou, and P. Hasapis, "Strategies for contextual reasoning with conflicts in ambient intelligence," *Knowl. Inf. Syst.*, vol. 27, no. 1, pp. 45–84, 2011, doi: 10.1007/s10115-010-0293-0.

[27] T. R. M. Braga Silva, L. B. Ruiz, and A. A. F. Loureiro, "How to conciliate conflicting users' interests for different collective, ubiquitous and context-aware applications?," in *Proc. IEEE Local Com. Netw. Conf.*, 2010, pp. 288–291, doi: 10.1109/LCN.2010.5735724.

[28] T. R. M. Braga Silva, L. Ruiz, and A. Loureiro, "Conflicts treatment for ubiquitous collective and context-aware applications," *J. Appl. Comput. Res.*, vol. 1, no. 1, pp. 33–47, 2011, doi: 10.4013/jacr.2011.11.03.

[29] M. Zoller and A. A. Nazari Shirehjini, "Realization of a UPnP agent for controlling adaptive multimedia environments," M.S. Thesis, Dept. Comput. Sci., Technische Universität, Darmstadt, Germany, 2008.

[30] S. Pizziol, C. Tessier, and F. Dehais, "Petri net-based modelling of human–automation conflicts in aviation," *Ergonomics*, vol. 57, no. 3, pp. 319–331, Mar. 2014, doi: 10.1080/00140139.2013.877597.

[31] N. Hanssens, A. Kulkarni, R. Tuchinda, and T. Horton, "Building agent-based intelligent workspaces," in *Proc. Int. Conf. Internet Comput.*, 2002, pp. 675–681.

[32] S. Pizziol, C. Tessier, and F. Dehais, "What the heck is it doing? better understanding human-machine conflicts through models," *CEUR Workshop Proc.*, vol. 885, pp. 44–49, 2012.

[33] F. J. Miandashti, M. Izadi, A. A. N. Shirehjini, and S. Shirmohammadi, "Conflict scenarios in smart homes," *IEEE Dataport*, 2020. [Online]. Available: https://dx.doi.org/10.21227/7rmk-6y51. Accessed: Aug. 1, 2020.

[34] S. Jiang and R. C. Arkin, "Mixed-initiative human-robot interaction: Definition, taxonomy, and survey," in *Proc. IEEE Int. Conf. Systems, Man, Cybern.* 2015, pp. 954–961, doi: 10.1109/SMC.2015.174.

[35] W. M. Trochim, "Hindsight is 20/20: Reflections on the evolution of concept mapping," *Eval. Program Planning*, vol. 60, pp. 176–185, 2017, doi: 10.1016/j.evalprogplan.2016.08.009.

[36] S. R. Rosas and M. Kane, "Quality and rigor of the concept mapping methodology: A pooled study analysis," *Eval. Program Planning*, vol. 35, no. 2, pp. 236–245, 2012, doi: 10.1016/j.evalprogplan.2011.10.003.

[37] S. Wutzke, N. Roberts, C. Willis, A. Best, A. Wilson, and W. Trochim, "Setting strategy for system change: Using concept mapping to prioritise national action for chronic disease prevention," *Health Res. Policy Syst.*, vol. 15, no. 1, pp. 1–13, 2017, doi: 10.1186/s12961-017-0231-7.

[38] R. Soellner, N. Lenartz, and G. Rudinger, "Concept mapping as an approach for expert-guided model building: The example of health literacy," *Eval. Program Planning*, vol. 60, pp. 245–253, 2017, doi: 10.1016/j.evalprogplan.2016.10.007.

[39] J. P. Donnelly, "A systematic review of concept mapping dissertations," *Eval. Program Planning*, vol. 60, pp. 186–193, 2017, doi: 10.1016/j.evalprogplan.2016.08.010.

[40] L. M. Vaughn, J. R. Jones, E. Booth, and J. G. Burke, "Concept mapping methodology and community-engaged research: A perfect pairing," *Eval. Program Planning*, vol. 60, pp. 229–237, 2017, doi: 10.1016/j.evalprogplan.2016.08.013.

[41] S. Alawneh, C. Howell, and M. Richard, "Fast quadratic discriminant analysis using GPGPU for sea ice forecasting," in *Proc. IEEE 17th Int. Conf. High Perform. Comput. Commun.*, 2015, pp. 1585–1590, doi: 10.1109/HPCC-CSS-ICESS.2015.26.

[42] Y. Yan, G. Liu, E. Ricci, and N. Sebe, "Multi-task linear discriminant analysis for View Invariant Action Recognition," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5599–5611, Dec. 2014, doi: 10.1109/ICIP.2013.6738585.

[43] A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim, "Human activity recognition via an accelerometer-enabled-smartphone using Kernel Discriminant Analysis," in *Proc. 5th Int. Conf. Future Inf. Technol.*, 2010, pp. 1–6, doi: 10.1109/FUTURETECH.2010.5482729.

[44] D. Tao, L. Jin, Y. Wang, and X. Li, "Rank preserving discriminant analysis for human behavior recognition on wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 813–823, Feb. 2014, doi: 10.1109/TII.2013.2255061.

[45] A. Iosifidis, A. Tefas, and I. Pitas, "Class-specific reference discriminant analysis with application in human behavior analysis," *IEEE Trans. Human-Mach. Syst*., vol. 45, no. 3, pp. 315–326, Jun. 2015, doi: 10.1109/THMS.2014.2379274.

[46] M. H. Siddiqi, R. Ali, A. M. Khan, Y. T. Park, and S. Lee, "Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields," *IEEE Trans. Image Process*., vol. 24, no. 4, pp. 1386–1398, Apr. 2015, doi: 10.1109/TIP.2015.2405346.

[47] A. Al Farooq, E. Al-shaer, T. Moyer, and K. Kant, "IoTC 2 : A formal method approach for detecting conflicts in large scale IoT systems," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage.*, 2019, pp. 442–447.

[48] V. Tuttlies, G. Schiele, and C. Becker, "Comity-conflict avoidance in pervasive computing environments," in *Proc. OTM Confederated Int. Conf. On Move Meaningful Internet Syst.*, Nov. 2007, vol. 4804, pp. 763–772, doi: 10.1007/978-3-540-76843-2.

[49] F. Dehais, M. Causse, F. Vachon, and S. Tremblay, "Cognitive conflict in human-automation interactions: A psychophysiological study," *Appl. Ergon*., vol. 43, no. 3, pp. 588–595, 2012, doi: 10.1016/j.apergo.2011.09.004.

[50] S. Resendes, P. Carreira, and A. C. Santos, "Conflict detection and resolution in home and building automation systems: A literature review," *J. Ambient Intell. Humaniz. Comput*., vol. 5, no. 5, pp. 699–715, 2014, doi: 10.1007/s12652-013-0184-9.

[51] D. Retkowitz and S. Kulle, "Dependency management in smart homes," in *Proc. IFIP Int. Conf. Distrib. Appl. Interoperable Syst.*, 2009, vol. 5523, pp. 143–156.

[52] C. Shin and W. Woo, "Conflict management for media services by exploiting service profile and user preference," *UbiComp Workshop (UbiPCMM)*, vol. 149, pp. 48–57, May 2005.

[53] S. Munir and J. Stankovic, "DepSys: Dependency aware integration of cyber-physical systems for smart homes," in *Proc. ACM/IEEE 5th Int. Conf. Cyber-Phys. Syst.*, 2014, pp. 127–138.

[54] J. Bohn, V. Coroama, M. Langheinrich, and M. Mattern, "Social, economic, and ethical implications of ambient intelligence and ubiquitous computing," in *Ambient Intelligence*, vol. 10, Berlin, Germany: Springer, 2005.

[55] G. Adamides, G. Christou, C. Katsanos, M. Xenos, and T. Hadzilacos, "Usability guidelines for the design of robot teleoperation: A taxonomy," *IEEE Trans. Human-Mach. Syst*., vol. 45, no. 2, pp. 256–262, Apr. 2015, doi: 10.1109/THMS.2014.2371048.

[56] M. A. Bujang and N. Baharum, "Guidelines of the minimum sample size requirements for Cohen's Kappa," *Epidemiol. Biostat. Public Health*, vol. 14, no. 2, pp. e12267-1–e12267-10, 2017, doi: 10.2427/12267.

[57] N. McDonald, S. Schoenebeck, and A. Forte, "Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice," in *Proc. ACM Human-Comput. Interact.*, 2019, vol. 3, pp. 1–23, doi: 10.1145/3359174.

[58] M. Lintean, V. Rus, and R. Azevedo, "Automatic detection of student mental models based on natural language student input during metacognitive skill training," *Int. J. Artif. Intell. Educ*., vol. 21, no. 3, pp. 169–190, 2012, doi: 10.3233/JAI-2012-022.

[59] T. S. Andre, and M. A. Schurig, "Teaching usability methods to undergraduate students: Can we measure changes in experience?," in *Proc. Hum. Factors Ergon. Soc*., vol. 51, no. 7, pp. 526–530, 2007, doi: 10.1177/154193120705100704.

[60] K. M. Carley, "Extracting team mental models through textual analysis," *J. Organ. Behav. Int. J. Ind. Occupat. Org. Psychol. Behav.*, vol. 18, pp. 533–558, 1997.

[61] W. M. Trochim and D. McLinden, "Introduction to a special issue on concept mapping," *Eval. Program Planning*, vol. 60, pp. 166–175, 2017, doi: 10.1016/j.evalprogplan.2016.10.006.

[62] J. De Leeuw and P. Mair, "Multidimensional scaling using majorization: SMACOF in R," in *Proc. Int. conf. Corresp. Anal. Related Methods*, 2011, pp. 1–15.

[63] I. Borg, P. J. F. Groenen, and P. Mair, "Purpose of MDS & Unfolding," in *Applied Multidimensional Scaling and Unfolding*, New York, NY, USA: Springer, 2018, pp. 31–34.

[64] K. M. Jackson and W. M. K. Trochim, "Concept mapping as an alternative approach for the analysis of open-ended survey responses," *Org. Res. Methods*, vol. 5, no. 4, pp. 307–336, 2002, doi: 10.1177/109442802237114.

[65] M. L. McHugh, "Lessons in biostatistics interrater reliability: The kappa statistic," *Biochem. Medica*, vol. 22, no. 3, pp. 276–282, 2012.

[66] I. Camacho, R. Carreira, P. Lynce and S. Resendes, "An ontology-based approach to conflict resolution in home and building automation systems," *Expert Syst. Appl*., vol. 41, no. 14, pp. 6161–6173, 2014, doi: 10.1016/j.eswa.2014.04.017.