TupperwareEarth: Bringing Intelligent User Assistance to the "Internet of Kitchen Things"

Sangjun Eom, *Graduate Student Member, IEEE*, Haozhe Zhou,
Upinder Kaur, *Graduate Student Member, IEEE*, Richard Voyles, *Fellow, IEEE*, and David Kusuma

***Abstract*—Smart devices have entered all spheres of modern living, from monitoring the steps we walk to managing refrigerator inventory, ushering in the dawn of a new urban experience. The kitchen is the heart of the home; a place to share, care for and nurture the family unit, but also a place seeing the greatest impact from the introduction of smart devices. The smart sensing and remote control capability of smart appliances have enabled great physical convenience for users but have had less impact on cognitive conveniences. While such devices can sense *what* they are working with, they fail to understand *who* they are working for, leaving much of the burden of trivial planning and decision-making to humans with less personalized services. Hence, we introduce TupperwareEarth, a knowledge-based ontological semantic network for the "Internet of Kitchen Things" with the aim of reducing physical as well as cognitive loads of humans in cooking tasks. Also, we present a testbed for exploring kitchen innovation and validating the effectiveness of TupperwareEarth that combines intelligent kitchen storage containers, Smart Tupperware, and existing smart kitchen appliances through an IoT network and a user-friendly front-end interface, Tuppy. Using this testbed, the quantitative user studies show a 33% reduction in average food preparation time and qualitative user surveys show that 75% of the users observed a significant reduction in cognitive loads, thereby validating the cognitive conveniences granted by TupperwareEarth.**

***Index Terms*—Internet of Kitchen Things, Smart Kitchen, Smart Tupperware, Ontological Semantics**

I. Introduction

Today, smart appliances such as smart speakers, smart security cameras, and smart thermostats have become quintessential parts of the modern home, bringing remote-control convenience to users anywhere on Earth. While the grand vision of a transformed ever-connected lifestyle, empowered by the Internet of Things (IoT) [1], has yet to be fully actualized, the increasing "interconnectedness" among devices and the ability of control from anywhere is valuable and welcomed.

Remote-control affords user conveniences that reduce physical burdens; turning lights on and off with a voice command through a smart speaker saves the time and energy of walking up to the switch. However, remote control has clear limitations when it comes to the kitchen, a space where more complex and sophisticated activities occur through interactions among the users and appliances. Kitchen activities, such as preparing meals, require further complex human planning and add cognitive burdens, as compared to the activity of turning the light on and off. While it is easier to automate push-button tasks, alleviating cognitive burdens demands intelligent decision making, that is customized to the user's preferences.

The kitchen requires extensive decision making from the user. The user decides what to cook based on factors such as availability of ingredients, personal preference, time of day, and the occasion. Further, the user must then orchestrate the process of cooking, from pre-processing the ingredients to using the array of devices needed for specific cooking techniques, finally cleaning and serving. The human-driven nature of activities in the kitchen demands smart devices to possess intelligence and an inherent comprehension of the person and the context to be of help. Yet, currently available smart devices just manage to automate certain physical processes such as heating for a certain time based on a pre-programmed recipe (for example, cooking potatoes in a microwave by just selecting the potato option). The limitation of such devices lies in their failure to predict the user's action to collaborate seamlessly. Hence, with the aim of addressing both physical and cognitive burdens of users in the kitchen, we present TupperwareEarth.

TupperwareEarth is an intelligent network of smart devices with the capability of "understanding" the user, the context of the kitchen, and decision-making abilities by leveraging a knowledge-based ontological semantics. The novel ecosystem of TupperwareEarth goes beyond push-button automation and alleviates cognitive burdens by integrating physical components with efficient communication network and intelligence. The Smart Tupperware acts as the physical layer of the network, connected using the communication and network layer, as illustrated in Fig. 1. The data aggregated from this network is processed using the analytic layer wherein the ontology-based knowledge is used to extract actionable items with respect to the context. By understanding the context of the activities and the availability of devices and ingredients, the system recommends appropriate recipe choices to the user. The user-friendly voice assistant enabled by the mobile application, Tuppy, acts as a bridge between the user and the network. The system guides the user throughout the process of each cooking task, thereby reducing both physical loads – by automating

This work was sponsored in part by the NSF under grants CNS-1726865 and CNS-1439717 with additional support from the NSF Center for Robots and Sensors for the Human well-Being (RoSe-HUB).

S. Eom is with Department of Electrical and Computer Engineering, Duke University, Durham, NC 27701, USA sangjun.eom@duke.edu

U. Kaur and R. Voyles are with Purdue Polytechnic Institute, Purdue University, West Lafayette, IN 47906, USA {ukaur, rvoyles}@purdue.edu

H. Zhou is with the Department of Computer Science, Purdue University, West Lafayette, IN 47906, USA zhou929@purdue.edu

D. Kusuma is with the Research and Product Innovation, Tupperware Brands Corps., Orlando, FL 32837, USA

the actuation of connected devices – and cognitive loads – of deciding next steps.

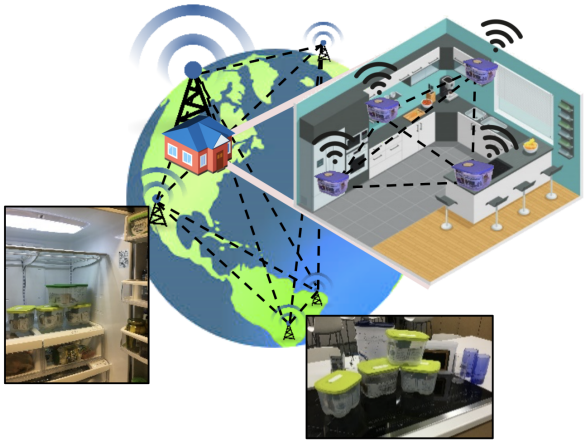


Fig. 1: A worldwide network of Smart Tupperware connected via TupperwareEarth to share data between containers for enhanced user experience and reduced burdens in the kitchen.

The three main contributions of this paper are listed as follows:

- A IoT-based network of Smart Tupperware that enables real-time monitoring and management of kitchen inventory through smart sensing, knowledge-based decision making and actuation.
- An ontology-based knowledge database of the kitchen for TupperwareEarth. An application of ontological semantics using this knowledge base is developed for the applications in suggesting recipes based on user preference, current inventory, availability of appliances, and other contextual information.
- Creation and validation of TupperwareEarth using the Smart Tupperware IoT network as a testbed. User study experiments go beyond physical load reduction to validate the effectiveness of TupperwareEarth in reducing cognitive loads in daily kitchen activities.

The organization of this paper is as follows: Section II presents the review of the published literature. Section III discusses the architecture of TupperwareEarth which consists of three layers: the physical layer of the system, the network layer, and lastly the analytics layer. The experimental validations are presented in Section IV with the results of the user study experiments. Finally, the discussion and the conclusion are presented in Section V and VI correspondingly.

II. Literature Review

Kitchen is the focal point of a household wherein people relax, seek nourishment, spend time as families preparing meals, and express their creativity through food. Being the "nerve center", the kitchen affects everyone, and this cuts across cultures and nationalities. A smart home's advanced kitchen is the one space that, arguably, has the greatest potential to enhance everyday lives.

Smart devices have the greatest potential to enhance user conveniences in the kitchen due to their existing intricate intertwinement of appliances in many diverse activities. Today, one can easily picture remotely starting the coffeemaker using a phone-based app or preheating the oven through a smart speaker by voice control. The remoteness in using these appliances brings conveniences, however, they are limited to a surface level. In order to ensure much deeper penetration of these conveniences in sophisticated and intricate activities, a richer understanding of the theater of the kitchen is needed.

In this section, the review of recent literature pertaining to the advancements in the integration of IoT-enabled automation in the modern kitchen is presented. The aim is to identify the shift of the vision in the kitchen IoT across the academic and industrial perspectives. The industry-driven goal of aggregating sensor data for enhancing productivity highlights the need for higher-level inference to transform the aggregated data into increased customer convenience. For this need, a review of the literature of semantics and existing ontology framework, in domains across IoT and food science, investigating the state-of-the-art is also presented in this section.

A. Kitchen IoT

The term "IoT" originated within the context of supply chain management [2], but the concept has rapidly extended across different applications, including homes, healthcare, transportation, and automation. In fact, one of the most significant growth areas of IoT integration is in our homes and kitchens. Not only for personal use, but with the rise of commercial services [3] such as Air-BnB (which allows homeowners to rent out their own homes), IoT-controlled locks, thermostats, and smoke alarms have become quite valuable – though, perhaps, not indispensable – to remotely manage the properties from a business case perspective. In this section, we review recent works in kitchen IoT and summarize our findings in Table I.

Prior works attempted integrating technology into the kitchen to bring intrinsic changes to daily kitchen activities. La Cantina [4] was a smart kitchen space from MIT where an interactive recipe was projected down onto the kitchen counter and allowed the user to manipulate over a capacitive touch sensor embedded on the surface. Living Cookbook developed by Terrenghi et al. [5] used a touch screen embedded on top of the stove to provide active cooking supports such as the location of tools, ingredients, and recipes. Similarly, Semantic Cookbook [6] recorded cooking processes and provided a software framework to share the records with other people to enhance the cooking activities. These early academic works shared the grand vision, with a rise of the IoT in the 2000s, of embedding technological advancements in various aspects of everyday life [17].

This grand vision of the IoT was perceived by both academia and industry, with the common thread of convergence of technology bringing positive impacts to daily operations, whether that be in the workspace or in the home. Several industrial products shown in Table I such as PantryChic [14]

TABLE I: The Review of the Kitchen IoT Literature

Author(s) / Name	Ingredient Recognition	Volume Detection	Expiration Warning	Recipe Suggestion	Cooking Instruction	Cooking State Detection	Appliance Automation	Personalized Adaptation
G. Bell & J. Kaye [4]					✓			
Terrenghi et al. [5]					✓			
M. Schneider. [6]					✓	✓		
Stander et al. [7]	✓				✓	✓	✓	
Blasco et al. [8]			✓				✓	✓
Gullà et al. [9]					✓			✓
Neumann et al. [10]					✓	✓	✓	
Achary et al. [11]		✓						
Shariff et al. [12]		✓	✓					
Jinila et al. [13]		✓	✓					
PantryChic [14]		✓		✓				
Ovie Smartware [15]			✓	✓				
Hestan Cue [16]					✓			
TupperwareEarth	✓	✓	✓	✓	✓	✓	✓	✓

and Ovie Smarterware [15] have emerged aiming to assist food pantry management. Smart cookwares with temperature sensors and wireless communication are also not rare [16]. The perception in academia mirrors the trajectory of industry-based IoT, which is the primary economic driver. Numerous innovative uses of sensors have enabled automatic monitoring of remaining volumes [11]–[13] and expiration date or spoilage of food ingredients [12], [13]. While both industrial and academic applications of integrating IoT into the kitchen, as described above, bring convenience to the user to some extent, they lack the capability to understand the data and the task, thereby still need user supervision albeit remotely.

To further unleash the full benefits of the data collected by these devices, several recent works focused on making inference through all devices in the kitchen environment instead of analyzing the data of each device separately. By augmenting user profile into the analysis, [8] and [9] are able to make adaptations to users of various types of impairments, facilitating their independent living. Other systems focus on interactive and intelligent cooking guidance. In [7] and [10], multiple sensors and cameras are used to detect the user's action and the state of the cooking environment, which enabled their systems to automate of the cooking appliance based on the detection in addition to simple remote control. In [7] an RFID sensor on their refrigerator enables detection of items with RFID tags. Through modeling the cooking workflow using XPDL4USE [18], a semantically annotated workflow language, the cooking guidance in [7] can automate kitchen appliances and transit upon completion of the previous step.

B. Ontological Semantics

Achieving a system capable of higher-level inference has been a topic of extensive research. The earlier work from Berners-Lee et al. [19], who invented the Web, sets a foundation for today's standards and formats of the ontologies with the development of OWL (Ontology Web Language) and RDF (Resource Description Framework) schema. In this literature review, the on-going work of integrating semantics through the ontologies in various domains has been described.

Ontology is about the nature of being that sets the logic and rules of how a particular "world" works. By containing information about how the nature of the kitchen works, ontology can provide knowledge to the system for reasoning and inferring the daily activities occurring in the kitchen sphere monitored by the IoT system. The ways of integrating an ontology into an IoT system vary in different approaches of addressing the integration at various layers of the hierarchy or through middleware or services. However, many of the ontologies described in the literature are still at the prototype stage and are specific to the applications of that particular research project [20].

IoT-Lite was proposed as a lightweight semantics engine of the Semantic Sensor Network (SSN) that reduced the time of query/response and simplified the annotations of the objects. However, the shallow depth of knowledge of the model is more suitable for sensor discovery with strict latency requirement than complex reasoning tasks with rich knowledge about kitchen and cooking [21]. FIESTA-IoT ontology aimed to address the interoperability issues by interconnecting existing IoT solutions of M3-Lite, SSN, and IoT-Lite [22]. The model facilitated interoperability by linking sevevral popular IoT ontologies; however, it created an issue of redundant information and lacked the support for context-aware services and information [23]. On the other hand, BOnSAI (Smart Building Ontology for Ambient Intelligence) is an ontology with a focus on services and context-awareness in the application of Smart Buildings [24]. Despite the number of ontologies in the IoT domain, an all-encompassing ontology for the kitchen is difficult to be found due to the broadness and complexity of the domain.

III. TupperwareEarth Architecture

In this work, we present a system that integrates advanced ontological semantics with the sensing and communication ability of smart containers for improving user experience, called TupperwareEarth. TupperwareEarth is an extended testbed for the "Internet of Kitchen Things" integrated with an ontological reasoning system that reduces

human planning burden through intelligent automation of the kitchen appliances and tasks. TupperwareEarth derives its name from "RoboEarth", a worldwide data and intelligence sharing network of robots [25]. Extending the concept of sharing data, TupperwareEarth uses ontological reasoning to add intelligence into the network of kitchen things as shown in Fig. 2. It enables communication among the Smart Tupperware containers and other kitchen appliances in the testbed, allowing them to operate together to accomplish complicated kitchen activities with the knowledge derived from the ontological database. Overall, TupperwareEarth as shown in Fig. 3, comprises of three major layers: 1) Physical Layer (constituting the devices and hardware), 2) Network Layer (constituting of the communication and networking infrastructure), and 3) Analytics Layer, implemented across different Amazon Web Service (AWS) platforms that offer serverless computing.

The physical layer of TupperwareEarth comprises the testbeds for the Internet of Kitchen Things. The testbeds include the in-lab created Smart Tupperware and other industrial kitchen appliances. Over the decade-long evolution of reformation in creating Smart Tupperware in different forms and functionalities [26], [27], [28], a new hardware of Smart Tupperware embedded into Tupperware's SmartFridge [29] container is created with a low-power-managed custom PCB and multiple off-the-shelf sensors. In this project, the Paragon Smart Induction Cooktop from First Build [30] is used as a part of the testbeds with multiple Smart Tupperware containers.

The testbeds build an infrastructure of the kitchen things for the network layer to form extensive communication between each kitchen appliance. Tens of Smart Tupperware containers residing in the kitchen build an inventory management system that monitors the food contents inside the containers and provide the raw data to the analytics layer. By extending the communication with other kitchen appliances, TupperwareEarth is capable of collecting more raw data from sensors embedded in each appliance and enables more services through the actuation of diverse kitchen activities. The communication further expands to Tuppy, the custom-built Android app for the user interface to deliver the enhanced convenience directly to the user at the front-end through a voice service.

The network layers is a gateway for the appliances to share the data and allow the analytics layer to further processing of the raw data transformed into information for user convenience. The analytics layer contains an ontological knowledge database that stores the data collected from the physical layer and inference engine with defined semantic rules. By reasoning based on the stored data with semantic rules, TupperwareEarth enables the applications for user convenience in providing a recipe suggestion and automation of the cooking tasks. These applications aim to reduce the human planning in cooking activities for enhanced convenience that overcomes the limitation of remote control in today's smart kitchen appliances.

A. Physical Layer

1) Smart Tupperware Hardware Design: Smart Tupperware is an ecosystem of electronics-embedded kitchen containers. Each container consists of an electronic substrate that monitors local sensors and connects to a wireless network [26]. Moreover, it hosts a suite of software applications, such as recipe suggestions or automatic shopping lists, that provide capabilities helpful to the kitchen nerve center [28]. In the past, different prototypes were fabricated in various shapes and sizes, including LCD-mounted container to barcode-embedded plastic cups, that explored various design challenges of building Smart Tupperware [27].

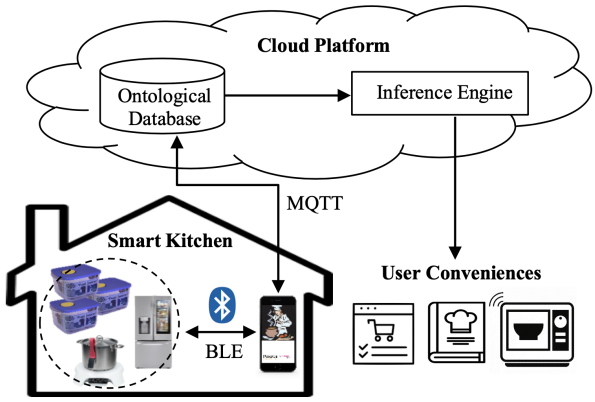


Fig. 2: The architecture of TupperwareEarth

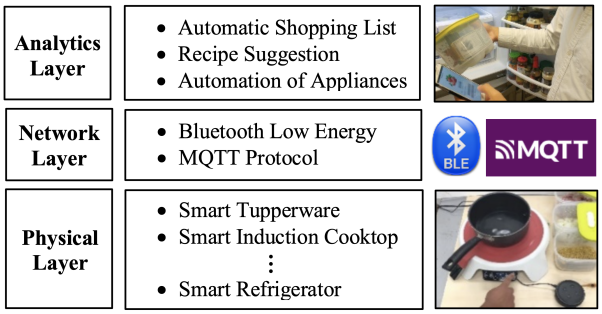


Fig. 3: The three layers of TupperwareEarth architecture with their essential functions.

Following up with the prior work on exploring different designs of Smart Tupperware, the new hardware was manufactured by incorporating the design challenges for low-cost and low-power systems. Tens of Tupperware containers, commonly used for food storage, are located everywhere in the kitchen from inside of a pantry to a refrigerator. Customers of the Tupperware containers do not desire expensive high-power electronics to be used as food storage everywhere in their kitchens. Thus, keeping these two design challenges as the major constraints for the microcontroller, size of the PCB, and number of sensors on-board were decided. The aim was to achieve low-cost manufacturing while maintaining low-power in operation. Using nRF52832 as a microcontroller with a built-in Bluetooth, the custom PCB minimized the number and cost of components as it excluded the external circuitry for wireless communication. The design specifications of Smart

TABLE II: The Comparison of Benchmark Specifications

	Smart Tupperware	**LCD Container** [26]	**Bar-code Cup** [27]	**Ovie Smarterware** [15]
Microcontroller	nRF5282	ATMega128L	-	-
Advertisement Current (mA)	~ 2	0.4	-	Unspecified
Battery (mAh)	80	950	-	Unspecified
Lifetime (days)	~ 365	~ 1460	Infinite	~ 365
Information Delivery	4.0" E-Ink Display	LCD Display	Embedded Bar-Code	LED Indicator
Sensors	Accelerometer, Light Intensity, Temperature, Humidity, Camera, Gas, Strain Gauge	IR, Strain Gauge, RGB Color	-	-
Wireless Communication	BLE	Bluetooth	RFID	BLE
Size of Electronic Substrate (mm)	59.8 x 107 x 2.85	279 x 94.0 x 170	73.7 x 73.7 x 145	9.53 x 31.8

Tupperware are shown in Table II and compared to the prior prototypes as well as the industrial application of Ovie Smarterware [15].

2) Mechanical Fabrication: While injection and over-molding are the conventional manufacturing techniques used in producing the plastic containers, these methods require a mold design that fits the machine, and survival of electronic substrate under these operations remains a challenge. Therefore, we first use a casting method with a curable liquid material by using a mold to duplicate the plastic container. The electronic substrate was placed inside the mold before the liquid two-parts polyurethane was poured, then cured. As a result, shown in Fig. 5a, the whole electronic substrate survived through the curing process and was tested for verifying that all functionalities were working correctly. The embedded container was able to update the E-Ink displays by connecting to the app via Bluetooth connection.

The same process was used to embed the electronic substrate into the form of the label to create a "smart label" which can be later over-molded onto the plastic container. The concept of over-molding a label comes from one of Tupperware's products that over-molds a thin printed label on the top surface. Embedding the electronic substrate by casting into the material had the advantage of exploring flexible designs. For instance, a two-split display design of electronic substrate was embedded, as shown in Fig. 5a, creating a smart label constrained within the allowed fit for the container. The overall dimension of the electronic substrate as shown in Figs. 4a and b are 59.79 mm by 107.0 mm which was under the required dimension of the mold for embedding the labels. The thickness of the electronic substrate was 2.85 mm as shown in Fig. 4c, which is much thicker than a conventional label embedded onto Tupperware containers but is under the maximum limitation of the mold dimension of 3.00 mm thickness.

Conventional manufacturing of plastic containers uses polypropylene that involves high temperature and pressure during the injection and over-molding processes. In comparison to the casting method of using polyurethane, which required relatively lower temperature and pressure for the curing process, the manufacturing of polypropylene containers makes it difficult for the electronic substrate to survive under harsh conditions. Although mechanical fabrication is not the focus of this paper, we conducted a number of trials of embedding electronic substrate into polypropylene through injection molding at Tupperware's production facility to explore the survival rate. The main challenge of successful molding was the fragility of the displays that were part of the electronic substrate. Notably, we found that the E-Ink displays were mostly compatible with a polypropylene under high temperature and pressure conditions as shown in Fig. 5b. However, the rigid-glass displays often did not survive the physical stresses of the process. We collaborated with Tupperware to explore preliminary work of mechanical fabrication in creating a fully fused form of Smart Tupperware, but more trials are needed to achieve a high success rate of fabrication.

3) Sensor Design: In the past, Smart Tupperware designs have explored various types of sensing such as weight, volume, and color of the food contents inside the container [26]. While the new hardware of Smart Tupperware inherits the past sensor designs, it adds six new sensors: accelerometer, light intensity, temperature, humidity, camera, and gas sensors. The set of new sensors allows each Smart Tupperware container to act as a rich data-aggregating IoT node, allowing the analytics layer to create more value out of the abundant raw data collected from each node. This value creation generates the desired information that enhances the user convenience and enables high-level inference at the front-end.

Smart Tupperware efficiently manages power by using an optimized wake-up interval for collecting sensor data and updating the display, while using minimal power for the rest of the time. Despite the one-hour interval sequence, there are occasions that demand Smart Tupperware to wake up due to sudden changes in the environment. For example, occasions when the user is actively using the container, so the display needs to provide the latest information or when the location of the container changes from inside of the refrigerator to the top of the kitchen counter with an uncontrolled temperature environment. All these situations require the system to recognize state changes such as change of usage or change of location of the containers. Once such a change is detected, the system updates the sensor data, correspondingly. We used LTR-329ALS for light intensity sensing and LIS3DHTR as the accelerometer to detect motion such as the opening of the lid or rigorous shaking of the container. Changes that exceed

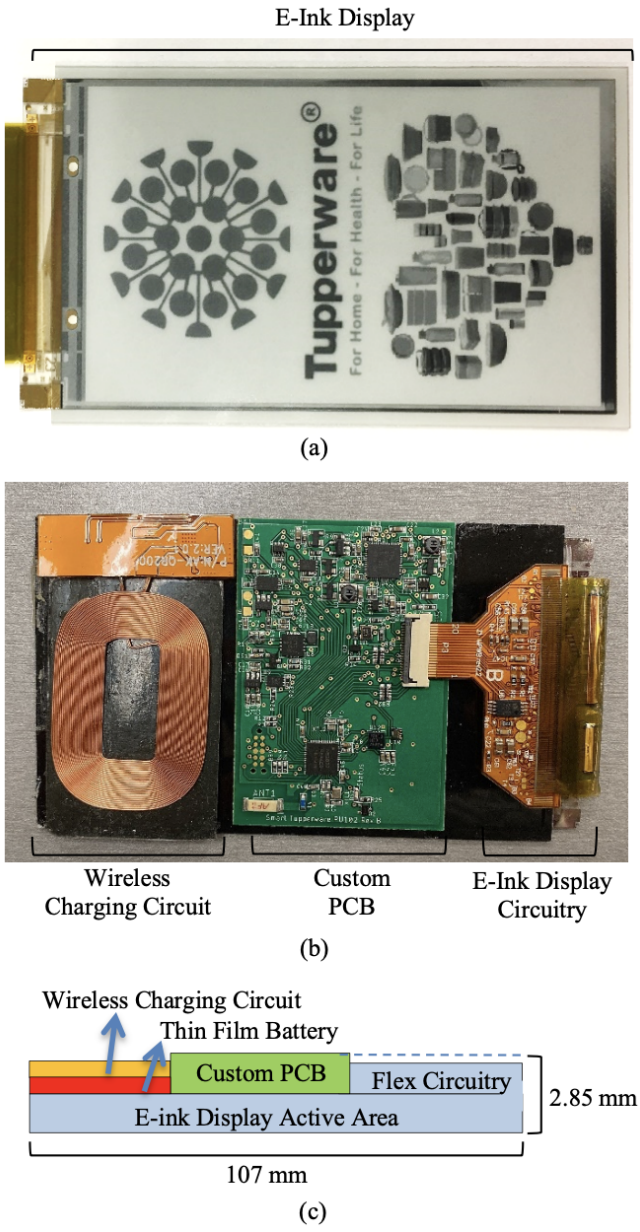


Fig. 4: The design of the electronic substrate used for Smart Tupperware: (a) the front incorporates the E-ink display for displaying information. (b) the back shows the wireless charging coils and the custom circuitry and (c) the thickness shows the height of different sub-components.

the threshold values for light intensity and accelerometer trigger the microcontroller to wake up, update sensor data and allow the users to control Smart Tupperware without manually turning the power on.

When tens and hundreds of Smart Tupperware containers connect together and collect data of the food stored in each container, an inventory management system is created in the kitchen that automatically monitors the conditions of all food contents. With the given set of sensors on board, Smart

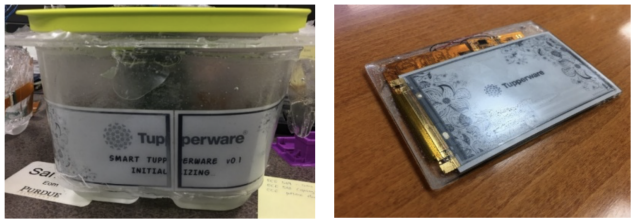
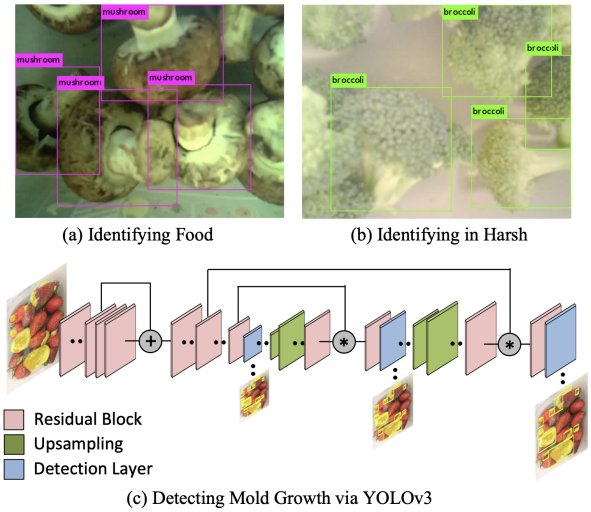


Fig. 5: The electronic substrate is embedded into the polyurethane by casting (a) into the container and (b) the label molds.

Tupperware is capable of providing numerical values of raw data as information to the user. However, one of the most desired information in the inventory management system is to identify the food type and automatically generate labels without human intervention. In the past, we have explored the designs of embedding bar-code labels into the cups to store the static information and allow the user to easily scan the codes through the plastics [27]. In our newest design, we added a camera sensor, OV2640 with a resolution of 1600 x 1200 pixels, to collect the images of the food contents and identify their type by using a real-time object detection model. Thus, we store dynamic information of food contents with automatic labeling.

To implement the object recognition system, we trained a YOLOv3 [31] using the images of the food contents collected from various Smart Tupperware containers. The training of identifying food type requires a massive database of pre-stored images of various food types. We finished a preliminary work of identifying two vegetables, mushroom and broccoli, in real-time as shown in Fig. 6a. Each 200 raw images were collected for mushroom and broccoli from the Smart Tupperware containers and processed for 80% and 20% split for train and test correspondingly. Pre-trained weights from the Imagenet were used and re-trained on the dataset for 10,000 iterations. The training accuracy resulted in 98.1% and the test accuracy was 94.7%. Some of the raw images collected from the containers used in this dataset had moisture and made the objects blurry as shown in Fig. 6b. This will be a common case when Smart Tupperware containers are used for a long-term monitoring of the food contents and further research on training in harsh conditions is recommended.

Another main burden that adds to the user's daily routine of managing kitchen inventory is checking on the expiration of food contents. Humans usually keep track of their inventories, remembering when they purchased the food contents and estimate the expiration dates. However, there are still occasions when the user directly needs to check whether the food contents expired or not by inspecting using sight, smell, touch, or even taste, if necessary. In the past, we explored a tracking system on the estimated expiration dates obtained from the product information, using bar-code or user's manual input to provide alerts to the user upon the expiration dates [28]. The



(a) Identifying Food (b) Identifying in Harsh

(c) Detecting Mold Growth via YOLOv3

Fig. 6: Real-Time Object Detection in Smart Tupperware using YOLOv3 for (a) identifying food types, (b) identifying food types in harsh conditions, and (c) detecting mold growth for food expiration.

new Smart Tupperware hardware adds three new environment sensors to collect humidity, CO2, and air quality levels and utilizes the camera sensor in detecting mold growth which is one of the most significant indicators of food expiration.

We used CCS811 as the gas sensor in measuring the Carbon Dioxide (CO2) levels in equivalent CO2 (eCO2) value and the air quality levels in Volatile Organic Compound (VOC) value. The humidity level is measured in relative humidity (RH) percentage by using a BME680 sensor. The combinations of these three measurements are used as the indication of the organic activities related to the level of freshness of the food contents. A threshold value was used for all three sensing parameters. Whenever the measurement levels exceeded the thresholds an alert was triggered.

Along with these measurements, detection of mold growth on food is added for a more accurate indication of a food expiration. We utilized the camera sensor using the YOLOv3 object detection again to train the system in detecting mold growth. Preliminary work on detecting the area of mold growth in strawberries are shown in Fig. 6c. We used 85 raw images collected from the containers with 70% to 30% split ratio of training to test. The training was completed with pre-trained weights from the Microsoft Common Objects in Context (COCO) dataset for 10 epochs. The accuracy on a training set was 96.67% and it was 92.00% on a test set. While mold detection also requires a large database of raw images, sharing the raw images between Smart Tupperware containers worldwide would help the training of the object detection and increase the accuracy of the identification. Both food type and food expiration detection are accomplished through sharing the same database of raw images saved in the cloud.

Fig. 7: The wireless communication between Tuppy and Smart Tupperware via BLE

B. Network Layer

The network layer connects the IoT devices in the physical layer to communicate with each other. It also acts as a gateway to exchange data with the analytics layer. Due to the large amount of raw data collected from each appliance, the network layer uses Bluetooth Low Energy (BLE) to keep power consumption low and also assigns a unique identifier for each physical device. On the other hand, a Message Queuing Telemetry Transport (MQTT) protocol is used to allow the analytics layer to subscribe and publish to a specific topic for reading and writing data from and to the kitchen appliances.

1) BLE Communication: BLE satisfies two requirements of our design challenges: keeping the power consumption of the electronic substrate low and enabling a bi-directional communication with a unique identifier for each device as shown in Fig. 7. These are advantages of BLE over past Bluetooth standards. We integrate each Smart Tupperware to be a unique IoT node in TupperwareEarth. Using BLE to wirelessly connect with an app draws about 2mA when the connection is made during the active mode.

2) MQTT Protocol: TupperwareEarth extends the connection of Smart Tupperware to the network of kitchen appliances through the MQTT protocol. The infrastructure of smart kitchen appliances exists, but interoperability remains a challenge with different types of communication and protocols being used. The wireless communication protocol varies from Bluetooth to Wi-Fi and the APIs are not shared as open-source by the manufacturers. MQTT brings an advantage of connecting the appliances under different protocols by exchanging data in a format of JavaScript Object Notation (JSON).

MQTT protocol uses a subscribe/publish model that provides flexibility for any appliance to subscribe as a client to exchange the data. The attributes are the topics that contain values of the data to which the server and client can subscribe to read the stored data or write the newly collected data to. For instance, Smart Tupperware comes with one actuator, a display, and several sensors. Each actuator and sensor data

TABLE III: The Comparison of Ontologies in IoT Domain

Name	TupperwareEarth	SOSA [32]	IoT-Lite [21]	FIESTA-IoT [22]	BOnSAI [24]
Domain of Application	Cooking, tool, appliance, recipe, sensing, actuating	Sensing, actuation, sampling	Device, entity, service, resource	Device, entity, service, resource, observation	Context, service, hardware, functionality
Contribution	Enable appliance automation, recipe suggestion and other knowledge-based decisions in kitchen context	Providing a lightweight and flexible ontology of IoT devices	Based on SSN that reduced response time and simplified annotations	Extension of IoT-Lite, addressing interoperability in hardware	Categorizing services and enabling context awareness
Approach	Incorporating cooking domain knowledge; ontological implementation of MILK [33]; extracting knowledge from network of Smart Tupperware	Modularization through vertical and horizontal segmentation	Only defining the most frequent IoT concepts	Merging and reusing existing popular ontologies	Extending existing ontology to provide ambient intelligence

are expressed as attributes containing a specific value. The attribute for the color sensor would contain RGB data as a value. This provides simplicity and ease of sharing data with the system, thereby reducing the communication traffic by customizing the subscription of the appliances.

3) Tuppy and Skillset: Tuppy is a custom android app that communicates with multiple Smart Tupperware containers at the same time and delivers the desired information to the users. Tuppy connects to each Smart Tupperware via BLE and obtain data from the analytics layer through the Wi-Fi. The user interface of Tuppy inherits from its precursor [26] which focused on delivering the raw data directly to the user by displaying them. However, Tuppy extends the assistance by enabling a voice-service and connecting to Alexa with Tuppy Skillset to allow the user to access the information without opening the app. The voice-assistance allows Tuppy to provide step-by-step instructions during the cooking process, alerts based on expiration of food contents and actively listen for user requests through Alexa.

C. Analytics Layer

The world of a kitchen closely revolves around the functionalities of various appliances and their uses in various cooking tasks. Humans comprehend the purpose and appropriate usage of each appliance in cooking tasks through years of learning and practice. We integrate an ontological knowledge database to embed a similar level of comprehension in TupperwareEarth. The knowledge database enables the system to understand the context of the kitchen, wherein it comprehends the existing knowledge about the coherence among kitchen appliances and cooking tasks. To understand semantics of the cooking tasks, TupperwareEarth uses recipes annotated in Minimal Instruction Language for the Kitchen (MILK) [33], which concisely abstracts a recipe into instructions of actions, ingredients and tools. Using the knowledge stored in the database, TupperwareEarth further infers semantic knowledge with a rule-based reasoning architecture.

An ontology is a formal organization of information, in terms of properties, entities, and their relationships, thereby providing a holistic map of the domain. An ontology helps to reduce complexity while studying and modeling a field. With the ontological model defining entities and relations among them, rule-based reasoning is used to infer new knowledge for adding semantics into TupperwareEarth. The integration of ontology and reasoning allows TupperwareEarth to deliver user convenience in the forms of recipe suggestion and automation of appliances. Both tasks are a major part of the cognitive and physical loads in the process of meal planning and cooking. With ontology defined for the kitchen of a particular user, TupperwareEarth delivers customized conveniences for the user.

1) Ontology for the Kitchen: TupperwareEarth's ontology is the knowledge database of our smart kitchen environment represented in Web Ontology Language (OWL). We built the ontology of TupperwareEarth using Protégé 5.5 [34], a popular tool for ontology editing. In contrast to existing IoT ontologies shown in Table III, the ontology of TupperwareEarth not only covers sensors and actuators on IoT devices, but also integrates the cooking-related concepts such as recipe, food and cooking techniques. Among several IoT models of ontology, SOSA (Sensor, Observation, Sample, and Actuator) is used as a foundation for TupperwareEarth's IoT ontological model. SOSA is a flexible lightweight ontology that provides a fundamental framework of sensors and actuators [32]. The advantage of using SOSA lies in the ease and simplicity of the implementation of the ontological framework. As the SOSA ontology was built for covering a broad range of sensors and actuators, the implementation is easily extensible to cover the different types of IoT-integrated kitchen appliances for TupperwareEarth.

In addition, we integrated MILK [33] into TupperwareEarth. MILK is a semantic parsing language for cooking recipes based on a first-order logic. By mapping MILK into ontology classes and properties, TupperwareEarth's ontology can model recipes step by step, providing a basis for inferring higher-level knowledge to understand the cooking techniques in relation to the appliances.

The class hierarchy of an ontological model describes the components of the ontology in Description Logic (DL), wherein each class contains subclasses and inherits the properties of the parental class. Inheritance of properties in classes creates *is-a-subclass-of* relationships that describe the taxonomy of the kitchen environment. TupperwareEarth derives its core structure from the SOSA model [32], wherein domain-specific knowledge about the smart kitchen environment is

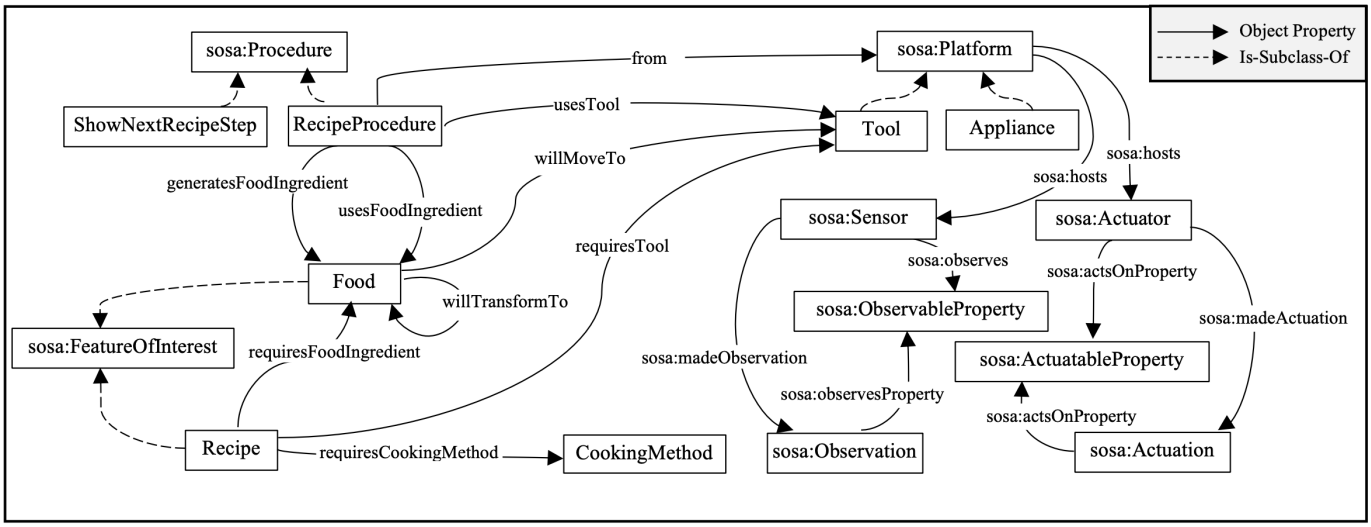


Fig. 8: TupperwareEarth's ontology structure showing various class hierarchies and object property

further integrated. While the SOSA model provides a definition of sampling, which enables measurements of a subset of the target, we do not include this class in the TupperwareEarth model as we want to keep the ontology lightweight. Hence, TupperwareEarth relies on one-time measurements from the sensors and actuators rather than taking repeated samples to reduce variance. The lightweight model thereby guarantees responsiveness as the inference time increases exponentially with complexity.

2) Class Hierarhcy: Fig. 8 shows the overall class hierarchy and object properties of TupperwareEarth. The TupperwareEarth ontology extends from the SOSA ontology [32] and maintains the majority of its concepts. Here we briefly explain the definition of necessary SOSA concepts used in TupperwareEarth. *sosa:Actuator* and *sosa:Sensor* are devices that performs an *sosa:Actuation* or an *sosa:Observavtion*. The target of *sosa:Actuation* or *sosa:Observavtion* is described by the class *sosa:FeatureOfInterest*. The characteristics of the *sosa:FeatureOfInterest* being observed or actuated are represented as *sosa:ActuatableProperty* and *sosa:ObservableProperty*. A *sosa:Procedure* is a reusable workflow that defines the specifications of *sosa:Observation* or *sosa:Actuation*.

The concepts in the device domain, shown in the right half of Fig. 8, largely followed the Observation-Actuation design inherited from SOSA ontology [32]. Since the kitchen activity has a strong temporal component, we add *madeLastObservation* as a subproperty of *sosa:madeObservation* to emphasize the latest observation result. An *Appliance* represents an IoT device connected to TupperwareEarth. The class *Tool* covers kitchen utensils, which is further divided into 13 subclasses according to the functionality of the *Tool* instance. For example, a Smart Tupperware is an *Appliance* and a *StorageContainer*, a subclass of *Tool*. The class *Appliance* is not subdivided into subclass to ensure the compatibility with smart appliances of diverse compositions and functionalities. We defined some subclasses within *sosa:Sensor* including *Camera*, *GasSensor*, *HumiditySensor*, *StrainGauge*, *Thermometer*, *Accelerometer* and *LightIntensitySensor*. We also defined some subclasses within *sosa:Actuator*, including *Display* and *Heater*. Furthermore we added six subclasses to *sosa:ObservableProperty* and three subclasses to *sosa:ActuatableProperty*. We used *sosa:hasSimpleResult* and *sosa:resultTime* to describe the result of *sosa:Actuation* and *sosa:Observation*.

While many concepts in the cooking domain are subclasses of *sosa:Procedure* and *sosa:FeatureOfInterest*, we defined new classes and object properties to map relationships among food ingredients, recipes, cooking actions, and entities in the device domain. Concepts in the cooking domain are shown in the left half of Fig. 8. The *Recipe* class and its associated classes *CookingMethod*, *Food*, *Tool*, are used to track the user's preference which enhances personalization of the recipe suggestion. The action in each step of the recipe is represented by *RecipeProcedure*. We created 14 such subclasses for *RecipeProcedure*, each for a distinct type of action in MILK [33]. At each step during a cooking session, a *RecipeProcedure* individual, along with its object property *usesFoodIngredient*, *generatesFoodIngredient*, *usesTool*, are created to denote the cooking instructions based on the MILK annotation from the recipe. Through ontology reasoning and incorporating with the information from the device domain, the object properties *willMoveTo*, *willTransformTo* and *from* are assigned automatically. These generated properties from reasoning not only enable accurate modeling of the entire cooking process, but are also essential for potential automation of kitchen appliances to provide the user with physical and cognitive convenience.

3) Rules for Reasoning: The logic of ontology is limited to DL, failing to provide high-level contextual information about the entities in terms of their properties. Thus, the reasoning rules have been integrated with the ontology to extract new knowledge based on the database and expand the use of the data. We create new associations in the form of specific

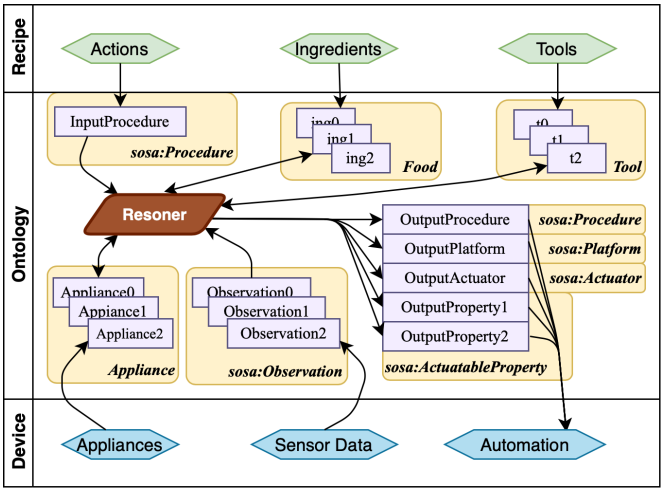


Fig. 9: The reasoning engine transfers data between ontology individuals, recipes and IoT devices and makes inference based on SWRL rules

TABLE IV: TupperwareEarth SWRL Rules for the Example Scenario

	SWRL Rule
①	madeLastObservation(?s, ?ob) ∧ sosa:hosts(?p, ?s) ∧ FoodType(?ft) ∧ Appliance(?p) ∧ sosa:Observation(?ob) ∧ sosa:Sensor(?s) ∧ sosa:observes(?s, ?ft) ∧ Food(?f) ∧ hasDescription(?f, ?res) ∧ sosa:hasSimpleResult(?ob, ?res) → sosa:hosts(?p, ?f)
②	Tool(?p) ∧ MILKProcedure(InputProcedure) ∧ usesFoodIngredient(InputProcedure, ?f) ∧ Food(?f) ∧ sosa:hosts(?p, ?f) → from(InputProcedure, ?p)
③	MILKCreateIng(InputProcedure) ∧ Tool(?p) ∧ from(InputProcedure, ?p) → ShowNextRecipeStep(OutputProcedure)
④	MILKCreateTool(InputProcedure) ∧ usesTool(InputProcedure, ?t) ∧ Stove(?t) ∧ Appliance(?p) ∧ Stove(?p) → sameAs(?t, ?p)
⑤	sosa:Actuator(?h) ∧ sosa:actsOnProperty(?h, ?temp) ∧ usesTool(InputProcedure, ?p) ∧ Appliance(?p) ∧ Timer(?b) ∧ Temperature(?temp) ∧ sosa:actsOnProperty(?h, ?timer) ∧ MILKCook(InputProcedure) ∧ sosa:hosts(?p, ?h) → sameAs(?timer, OutputProperty2) ∧ sameAs(?h, OutputActuator) ∧ sameAs(?p, OutputPlatform) ∧ sameAs(?temp, OutputProperty1)
⑥	generatesFoodIngredient(?p, ?y) ∧ usesFoodIngredient(?p, ?x) ∧ MILKProcedure(?p) ∧ Food(?x) ∧ Food(?y) → willTransformTo(?x, ?y)

rules created using Semantic Web Rule Language (SWRL), a language developed for the Semantic Web [19]. Creating rules in SWRL enables the system to infer new knowledge based on the existing knowledge stored in the ontological database by embedding richer associations. Furthermore, rule-based reasoning brings advantages in the form of memory savings and simplification of the taxonomy of ontology. Also, it provides the flexibility of adapting the user-desired reasoning with any inference engine [35].

The SWRL rules for TupperwareEarth construct logical reasoning in the following four domains: cooking technique inference, mapping of recipe entities and physical entities, kitchen environment modeling, and appliance automation. A complete list of the rules is shown in the Appendix section. Inferring cooking technique uses six rules that utilize the availability of *Tool* and historical data of cooked recipes. We assume that each *Tool* is capable of at least one cooking method. Five rules are used for the mapping between the entities in the recipe and the entities in the physical world. In this mapping, we assume that each tool and food ingredient in the recipe corresponds to a unique physical entity in the kitchen, thus representing a unique individual in the ontology. Four rules were defined to monitor the kitchen environment during cooking. These rules infer the new kitchen state when the cooking instruction and sensor data are provided. For example, if the current cooking procedure requires putting the ingredient stored in this Smart Tupperware to another *Tool*, the ontology will assert an inference that this procedure is already completed by the user when the weight sensor in Smart Tupperware observes a reduction of value. Automation of appliances uses five rules that utilize cooking instructions and available appliance information to determine which *Appliance*, with its associated *sosa:Actuator* and *ActuatableProperty*, is suitable for the cooking step.

The rule-based reasoning is controlled by our inference engine, which employs OWL API [36] and HermiT reasoner [37]. The inference engine is also responsible for the communication across recipes, IoT devices, and the ontology. Fig. 9 shows the data flow of the communication and the role of the SWRL rule-based reasoning. For each step in the recipe, the inference engine reads the annotated recipe, translates to assertions to the ontology, and launches HermiT reasoner. The new knowledge inferred by HermiT reasoner based on the SWRL rules are used for recipe suggestion and automation during cooking. The inference engine then performs necessary cleaning and updating based on the inference results before proceeding to the next step.

4) Example Scenario: We illustrate the use of SWRL rules in TupperwareEarth with the following example recipe snippet:

```
create_ing(ing0, "broccoli")
create_tool(t0, "stove")
cook(ing0, t0, ing1, "cooked broccoli", "")
```

The rules used for this example scenario are listed in Table IV. Assume we have the following precondition: A Smart Tupperware containing broccoli is available in the kitchen. The Smart Tupperware device is modeled as an individual *SmartTupperware0* of class *Appliance* and *StorageContainer*, a subclass of *Tool*. In addition, a smart stove is available in the kitchen and is modeled as an individual *SmartStove0* of class *Appliance* and *Stove*, a subclass of *Tool*, in the ontology. The *SmartStove0* has its own individuals of *Temperature* and *Timer*, which are subclasses of *sosa: ActuatableProperty*.

At the first step, the inference engine reads the recipe `create_ing`. Based on the inputs to the ontology, the inference engine first creates a new individual *ing0*, which *hasDescription* "broccoli", and at the same time asserts the class of *InputProcedure* to be *MILKCreateIng*. By rule ①, the reasoner infers that *SmartTupperware1 sosa:hosts ing0*.

By the rule ②, the reasoner asserts that the ingredient used by *InputProcedure* is *from SmartTupperware1*. Then by Rule ③, the reasoner asserts that *OutputProcedure* is of class *ShowNextRecipeStep*, meaning this procedure is complete because TupperwareEarth detected that the required ingredient is available. The inferred class type *ShowNextRecipeStep* is returned to the inference engine, which then proceeds to the next step in the recipe and cleans up the ontology by removing the inferred assertions of the input and output individuals.

At the second step, the inference engine reads the recipe `create_tool`, and creates a new individual *t0* of class *Stove*. It also asserts that the class of *InputProcedure* to be *MILKCreateTool*. By rule ④, the reasoner asserts that this *t0* is the same individual as *SmartStove0*. This means that we will use *SmartStove0* in the kitchen for the subsequent uses of *t0* in the recipe. The reasoner also asserts that the *OutputProcedure* is of class *ShowNextRecipeStep*. The inferred class *ShowNextRecipeStep* is, again, read by the Reasoning Engine to proceed to the next step and clean up temporary assertions in the ontology.

At the third step, the inference engine reads the recipe `cook`. The inference engine creates a new individual *ing1* that *hasDescription* "cooked broccoli". By the arguments of `cook`, inference engine adds these assertions: *InputProcedure is-a MILKCook*, *InputProcedure usesTool t0*, *InputProcedure usesFooodIngredient ing0*, *InputProcedure generatesFoodIngredient ing1*. Then by Rule ②, the reasoner asserts that *InputProcedure* takes its ingredient *from SmartTupperware0*. By Rule ⑤, the reasoner infers the output individuals should be the *SmartStove0* and its associated actuatable *Temperature* and *Timer* properties. The inferred types of the output individuals are read by the inference engine to upload the temperature and timer settings to the smart stove in the kitchen. At the end, by rule ⑥, *ing0 willTransformTo ing1*. After the user completes this step, the inference engine will remove *ing0*, and assert that *SmartStove0 sosa:hosts ing1*.

In the above example recipe, TupperwareEarth and the SWRL rules reduce human planning and intervention by: avoiding two steps of manual checking of kitchen environment, proceeding two steps in the recipe without interrupting the user, and automatically uploading temperature and timer settings to an appropriate cooking appliance.

5) Recipe Suggestion: Human planning in cooking activities in the kitchen involves both cognitive and physical loads. These loads are associated with activities that demand the user to have a clear thought-process in making decisions or to physically complete a task. While physical loads can be attributed to the physical performance of the task, such as cutting, lifting, and churning, etc., cognitive loads are attributed to planning, measuring, and checking the status of the tasks or the processes. The human planning associated with various cooking processes is complex and demanding, hence TupperwareEarth aims to reduce these loads by providing meaningful assistance. TupperwareEarth brings advanced conveniences to the user by sharing the cognitive loads as it suggests personalized recipes for each user, and also helps in the cooking process by automating various sub-processes and providing step-by-step assistance.

Choosing a recipe requires the user to not only consider what types of ingredients are currently available in the kitchen and what type of cooking techniques can be performed with available kitchen appliances, but also what types of cuisine or recipe category are preferred. However, many of the existing online recipe suggestion or searching systems such as Allrecipes [38] and Cookbooks [39] adds more cognitive loads and human planning by requiring the users to manually check the inventory and specify the preference. Using the inventory management system of Smart Tupperware and ontological reasoning, TupperwareEarth holds the advantage of reducing human planning from the stored knowledge about ingredients, kitchen appliances, and user profile.

To address the demand of the user comprehensively, our recipe suggestion adopts a decision-tree that utilizes the knowledge about cooking techniques and user preference from the ontology. The reasoning-integrated system first narrows down its search space to the recipes that only use cooking techniques that can be performed, given the availability of kitchen appliances as well as the user's historical cooking records. Then, the system infers the preferred recipe categories based on the user's demand. At the last step, the system calculates the scores for each recipe based on the sum of scores in matching, missing, and extra ingredients between the inventory and required ingredient lists. The scoring function is defined in Eqn. 1 shown below, where S_{miss}, S_{extra}, S_{match} are defined in Eqn. 2.

$$S(I_u, I_r) = \frac{S_{miss}(I_u, I_r) + S_{extra}(I_u, I_r) + S_{match}(I_u, I_r)}{|I_r|} \tag{1}$$

$$\begin{cases} S_{miss}(I_u, I_r) = m_1 \sum\limits_{i \in I_r \setminus I_u} w(i, c_r) \\ S_{extra}(I_u, I_r) = m_2 \sum\limits_{i \in I_u \setminus I_r} w(i, c_r) \\ S_{match}(I_u, I_r) = m_3 \sum\limits_{i \in I_u \cap I_r} w(i, c_r) \end{cases} \tag{2}$$

In Eqn. (2), I_r represents a list of required ingredients from the recipe and I_u represents a list of available ingredients in the user's kitchen inventory. We use $w(i, c_r)$, the relative frequency of ingredient i present in category of the recipe c_r, to weigh the ingredients. The intuition is that recipes within a category tend to share the same set of major ingredients that cannot be replaced for cooking. m_1, m_2, and m_3 are used to adjust the reward or penalty of missing, extra, and matching ingredients. In practice, we set $m_1 = -2$, $m_2 = -0.5$ and $m_3 = 2$.

Our recipe suggestion system is integrated with Tuppy to allow the user to request a suggestion and obtain the recipe as a result through the voice service. The user can either directly use the Tuppy app to request the in-app voice service or use the smart speaker to use Tuppy skillset. In both ways,

Tuppy delivers convenience to the user through interaction as a kitchen assistant. Due to a limited length of response and the inefficiency of listing multiple top-scored recipes, Tuppy delivers the highest scored recipe and leaves the rest as optional information to check on the app.

Automation of Appliances The remote control requires humans to initiate the action due to the lack of understanding of the intent from the system. The human initiation adds cognitive loads, but also physical loads if the appliances need the user to open an app and click a button without a voice service. While the convenience desired in the kitchen is not a full autonomy since the user still needs to drive the cooking actions such as stirring or picking the ingredients, TupperwareEarth uses the ontological database to infer which appliance needs to be automated for what type of cooking tasks.

Twenty SWRL rules in the TupperwareEarth ontology define the relations between the cooking tasks and functionalities of the kitchen appliances to infer knowledge about which appliance with what type of sensors and actuators can perform what type of cooking tasks. For instance, a cooking task of boiling pasta noodles can be inferred by the rules to automate the smart cooktop to set to boiling temperature with a timer. The automation of cooking tasks reduces the cognitive loads of the user initiative in setting up the temperature and timer on the smart cooktop by manually clicking buttons.

IV. Experimental Validations

Using the *TupperwareEarth* testbed, we designed the user study experiment with the goal of investigating the significance of TupperwareEarth's assistance–in the form of recipe suggestion, cooking guidance, and appliance automation–on the physical and cognitive loads of users. The testbed includes three Smart Tupperware containers, a Paragon smart induction cooktop, and an Amazon Echo smart speaker enabled with Tuppy as shown in Fig 10. In the experiment, each participant was given the same testbed to perform the task of cooking a macaroni salad twice; the conventional method is used in the first trial wherein the participant manually controls the appliance; the second trial uses TupperwareEarth to assist the user through the task. The meal preparation time, measured from the initial interactions with the testbed and ingredient till the completion of appliance setups for cooking pasta al dente, was compared between the participant groups in the two settings and further analyzed for the reduction in time, physical and cognitive loads.

Prior to the first trial, each participant filled out a pre-study questionnaire to answer questions related to their level of expertise in the cooking task and in the operation of smart kitchen appliances (SKA). Five different levels of expertise were given as choices for self-identification for each question. The participants were also asked to estimate the average time for them to decide on a menu based on a provided list of available ingredients and then locating the specific ingredients from their own kitchens. After the completion of the second trial, participants were asked to fill out the post-study questionnaire. Each participant rated the impact of the

(a) (b)

Fig. 10: The user study experiment setup with TupperwareEarth testbed (a) at the start of the meal preparation and (b) after the completion of the meal preparation.

TupperwareEarth system on their time-to-prepare and their physical and cognitive loads. Overall, a total of 16 participants completed the user study experiments.

A. Survey Results

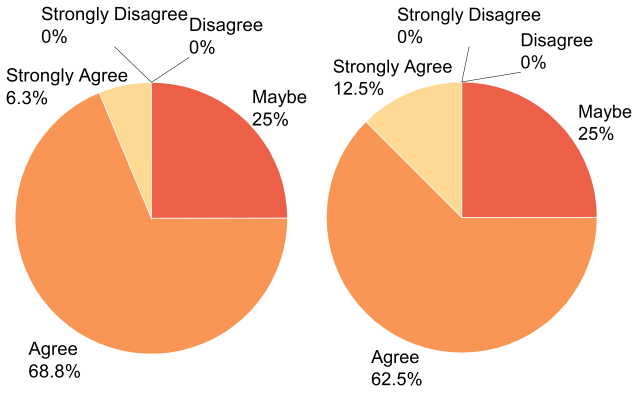
In the pre-study questionnaire, participants were asked to identify what they perceive as the sub-task with the highest cognitive load in the meal preparation and cooking process. Of the top three identified tasks, two of them which were "locating and knowing the list of ingredients" and "configuring the appliances." These tasks involve less physical loads, but require more cognitive loads of understanding and decision-making based on knowledge. The ontological system with the knowledge database enables TupperwareEarth to directly address these desired assistance by reducing both physical and cognitive loads of the tasks. However, the third identified task, "verification of the ingredient being cooked" is directly related to the quality of cooking and is not currently address by the system.

- 7 users identified "Locating & Knowing the list of ingredients"
- 5 users identified "Verifying that the ingredient is cooked"
- 2 users identified "Configuring the appliances"

Another question asked to the participants in the pre-study questionnaire was to identify the most desired assistance to receive from the smart kitchen appliances. The top four most desired assistance identified by the participants are listed below:

- Voice assistance for cooking instructions
- Automatic setup and timer, Alerting system
- Locating the ingredients
- Finding a recipe

These assistance require remote control to be a part of the functionality, however, they also require the system to make high-level inferences such as, when to alert the user, or making a customized recipe suggestion. TupperwareEarth addresses the all top four most desired assistance for enhancing the convenience and overcomes the limitation of just remote control in smart kitchen appliances.



(a) (b)

Fig. 11: Post-study questionnaire results from the User study experiments on TupperwareEarth's reduction of (a) cognitive loads and (b) perceived time in cooking tasks.

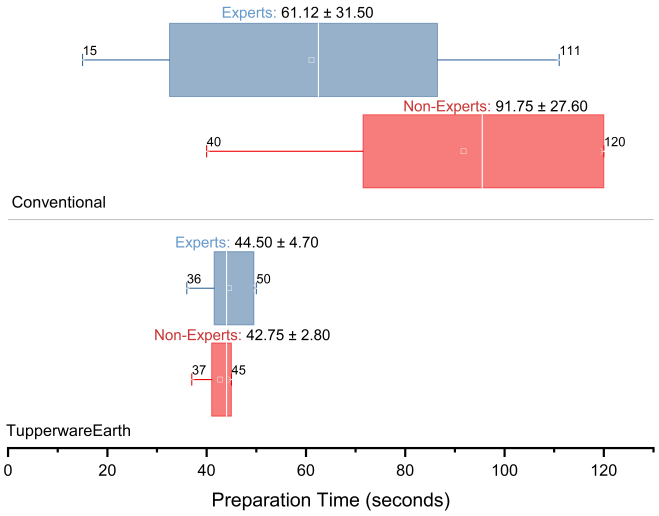


Fig. 12: Box plots for average preparation time between the conventional and TupperwareEarth methods by experts and non-expert groups in cooking skills.

In the post-study questionnaire, participants were asked the following two questions in regard to their interactions with TupperwareEarth system with answer choices of *strongly-agree, agree, neutral, disagree, strongly-disagree*:

- Did the assistance from TupperwareEarth reduce your time for the cooking task?
- Did the assistance from TupperwareEarth reduce your cognitive load from the cooking task?

Fig. 11, shows that about 75% of the participants gave positive answers by responding either "agree" or "strongly agree" to the reduction in both perceived time and cognitive loads of the cooking task. The high percentage of positive answers shows that TupperwareEarth is addressing the need of intelligent assistance in meal preparation from smart appliances in the kitchen.

B. Analysis

1) Average Preparation Time: In addition to collecting qualitative data from the survey results on perceived time and cognitive loads of meal preparation, we also analyzed the average meal preparation time between the groups of participants from different groups to investigate the impact of TupperwareEarth's assistance.

We used a two-sample paired t-test to compare the meal preparation time between the conventional method and the TupperwareEarth assisted method with Bonferroni correction. Table V summarizes the results of the analysis, including the mean decrease in time and the 95% confidence interval, where * and ** indicate p-values less than 0.05 and 0.01, respectively. Regardless of cooking and SKA expertise, the meal preparation time with assistance from TupperwareEarth is on average 32.81 seconds less than using conventional methods ($p = 0.0015$). A statistically significant decrease of time is also found in the group of cooking non-experts ($p = 0.0030$) and the group of SKA non-experts ($p = 0.0018$). While the effect of TupperwareEarth on meal preparation time is still in the positive for the two expert groups, the paired t-test showed no statistical significance with the Bonferroni correction, which can be attributed to the limited sample size of the participants in these groups.

The average preparation time for the expert and non-expert participants are shown in Fig. 12. Both participant groups lessened their time consumed in meal preparation with the TupperwareEarth assisted approach. The group of experts has a time reduction of 16.6 seconds, whereas the group of non-experts has a more substantial time reduction of 49.0 seconds, indicating a 53% reduction of average meal preparation time from the conventional method. Unlike the wide variations in performance observed in the conventional method, the preparation time with assistance from TupperwareEarth distributed within a small range of 40 to 46 seconds. Although the experts seemed to spend slightly more time than the non-experts when using TupperwareEarth, a two-sample t-test does not show any significance in this distinction ($p = 0.4110$). Consequently, the result suggests that TupperwareEarth not only saves time on meal preparation, but also mitigates the gap between cooking experts and beginners. The reduction in time is consistent with our goal of designing TupperwareEarth to reduce the burdens of the users.

TABLE V: The Comparison of Average Preparation Times

Participants Group	Average Preparation Time(s)		Decrease in Time(s)
	Conventional	Tupperware Earth	
Expert Cooking Skills	61.1	44.5	16.6 (-7.96, 41.22)*
Non-Expert Cooking Skills	91.8	42.8	49.0 (22.84, 75.12)**
Expert SKA	62.2	40.2	22.0 (-31.30, 75.30)
Non-Expert SKA	82.9	45.2	37.7 (17.71, 57.75)**
All Participants	76.44	43.63	32.81 (14.82, 50.80)**

2) Physical and Cognitive Loads: We further analyzed the average preparation time in terms of physical and cognitive loads by measuring an absolute physical load time and the cog-

nitive load time. In the experiment, we measured the absolute physical load time with a participant who is an expert for both cooking skill and operating the smart kitchen appliance. This participant is fully trained for the given cooking instructions in this experiment. The assumption is that the absolute time of physical load only involves the physical labor of each cooking tasks such as picking up the ingredient, placing the pot onto the cooktop or clicking buttons for the cooktop settings. Ultimately, it takes the same amount of time in physical loads across all participants. The average time of absolute physical load for the conventional method took 10.7 seconds, while the time increased to 19.2 seconds when the participant was assisted by the TupperwareEarth system. Subtracting the physical load time from the average preparation time yields the cognitive load time, as shown in Table VI.

The results show a significant reduction in cognitive loads from 63.8 seconds to 24.0 seconds when TupperwareEarth is used. The cognitive loads include understanding, planning and decision-making. For instance, the participants had to determine what ingredients were needed at which stage of the recipe, what temperature and timer settings were needed for the cooking skill, and the understanding of how to use the cooktop. The non-expert group in cooking skill benefited the most significant reduction in the time of cognitive loads by 57.5 seconds which consists of 71% reduction from the time of cognitive loads taken in the conventional method. The results from the experiment proves that the difference in level of expertise in cooking skill and operating smart kitchen appliances affected the time of cognitive loads in completing the cooking tasks.

However, the increase in the time of physical load shows a contrast to the reduction in cognitive loads. Although remote control reduces the burden of physically walking to the devices and controlling them, a new digitized way of operating the smart appliance using buttons and screens adds more time than a traditional way. For instance, a simple knob replaced by the buttons with touchscreen or pulling out an app to operate the appliance for remote-control in fact adds more physical loads (in terms of time) for the users. Therefore, remote control is not the most desired assistance as the new way of controlling smart appliances becomes an additional burden. The same trend was seen in TupperwareEarth's assistance by adding 8.2 seconds more to the physical loads. Most of the additional time came from the user's waiting time wherein they were waiting for the voice-alerts and guidance from the system. However the reduction in cognitive loads of 39.8 seconds was more significant than the addition of physical loads. As a result, the total preparation time still reduced.

The same absolute physical time is assumed in this experiment to separate the physical and cognitive loads out of the total meal preparation time. However, there are still variations among the users on performing the same physical tasks. People have different speed of action, such as grabbing the ingredients or stirring. They also make mistakes that can slow down the process of the physical tasks in cooking. Therefore, we can hypothesize that the difference in level of cooking skills or smart kitchen appliances can have impacts on the physical loads. The experiment conducted in this study was designed to measure the physical loads as the time for meal preparation, however a new study to investigate further about the variation of physical loads across the different participant groups is recommended.

TABLE VI: Physical and Cognitive Loads of Average Preparation Time

Method	Average Preparation Time(s)			
	Conventional		TupperwareEarth	
Time Consumption Category	Abs Physical	Cognitive	Abs Physical	Cognitive
Expert Cooking Skill	10.7	50.4	19.2	25.3
Non-Expert Cooking Skill		81.1		23.6
Expert SKA		51.5		21.0
Non-Expert SKA		72.2		26.0
All Participants	10.7	63.8	19.2	24.0

V. Discussion

TupperwareEarth introduces a new paradigm of customized conveniences for users in the kitchen. The Smart Tupperware hardware distinguishes itself from the ones previously published [15], [26], [27] by integrating sensors and actuators designed for enabling a wide range of applications, such as identifying food type and monitoring food expiration. With our new, enhanced fabrication method of embedding the electronic substrate into the plastics, we are able to deliver low-cost and low-power smart containers that are suitable for the kitchen environment. Smart Tupperware will evolve further to be dishwashable and food safe. This advantage will make Smart Tupperware ideal for large scale use and bring meaningful customization to the users. We aim to deliver a commercialized adaptation of Smart Tupperware in the future.

Seamless communication is the backbone of TupperwareEarth, tying all the pieces together. The integration of BLE and MQTT protocols allow for efficient data management among multiple IoT devices while maintaining low latency across applications. Leveraging the seamless communication among the devices, TupperwareEarth delivers step-by-step instructions of the cooking process in real-time through Tuppy. We prove that no statistically significant physical and cognitive burdens were added from this application. The adaptation of the industry standard communication protocols and intelligent voice-based control through Tuppy will further bring user incentives to our commercialized products. While we plan to expand the TupperwareEarth's IoT network to other kitchen appliances, such as smart refrigerator or smart microwave, to offer a greater set of services, we will also explore the security enhancement in our network to ensure a safe communication environment for each individual user.

The hardware and communication implementations are further enriched by their intertwining with the semantic architecture. The ontology-based semantic architecture grants comprehension of the tasks, the needs of the user, and the functionalities that each appliance offers. TupperwareEarth is the first of its kind to leverage multi-domain ontological semantics

with smart kitchen appliances to attempt to "understand" the user and the context of the kitchen by mapping the capabilities of the appliances to the needs of the task at hand. We plan to further enhance the customization by leveraging the users' shopping and usage data into TupperwareEarth's ontology, deriving individual user's daily patterns to personalize the recipe suggestions and automatic shopping list.

VI. Conclusion

The rise of Internet-of-Things has propelled the integration of technology in the kitchen. While numerous smart appliances exist today, they primarily allow remote control and monitoring leaving the human to still be in charge of the orchestration of the processes. TupperwareEarth extends the utility of smart devices by coupling them to an intelligent network that not only grants the ease of remote control but also alleviates the cognitive burdens of the user by actively participating in the planning and orchestration of the cooking process.

In this paper, we implement TupperwareEarth by using Smart Tupperware, a novel IoT-based smart container that enables real-time monitoring and management of kitchen inventory. These containers are connected by an agile network to the TupperwareEarth system. Using an ontology-based knowledge database for the kitchen, TupperwareEarth comprehends the availability of appliances and ingredients, and also understands user preference in the context of cuisines and time of day. The ontology provides the knowledge to suggest recipes to the user based on the inferred context. The TupperwareEarth system then guides the user through the process of cooking with the suggested recipe through the Tuppy user interface application.

The user study experiment proves that TupperwareEarth reduces average food preparation time for both expert and non-expert cooks. Notably, 75% of the users agreed that TupperwareEarth resulted in reduced cognitive loads and reduction in time of the meal preparation. Analysis of expert and non-expert participants (classified in terms of cooking skills and use of appliances) showed that both groups benefited from the use of TupperwareEarth. The group of non-expert users benefited the most from TupperwareEarth with a noted reduction of 53% of the total meal preparation time and 71% of the time in the cognitive loads. Hence, the user studies illustrate the effectiveness of TupperwareEarth in the kitchen. In the future, the scope of TupperwareEarth can be extended by incorporating robotic technology to assist the user in further sophisticated kitchen processes, cooking more intricate recipes.

APPENDIX

TABLE VII: List of SWRL Rules in TupperwareEarth Ontology

Catgory	Name	Condition		Inference
		Appliance	**Cooking**	
Cooking Method	1	DeepFrier(?o)		Frying(NextRecipeCookingMethod)
	2	Grill(?o)		Grilling(NextRecipeCookingMethod) ∧ Broiling(NextRecipeCookingMethod)
	3	MicrowaveOven(?o)		Microwaving(NextRecipeCookingMethod) ∧ Roasting(NextRecipeCookingMethod) ∧ Baking(NextRecipeCookingMethod) ∧
	4	Oven(?o)		Baking(NextRecipeCookingMethod) ∧ Roasting(NextRecipeCookingMethod)
	5	Stove(?o)		Boiling(NextRecipeCookingMethod) ∧ Blanching(NextRecipeCookingMethod) ∧ Braising(NextRecipeCookingMethod) ∧ StirFrying(NextRecipeCookingMethod) ∧ Brewing(NextRecipeCookingMethod) ∧ Frying(NextRecipeCookingMethod) ∧ Poaching(NextRecipeCookingMethod) ∧ Stewing(NextRecipeCookingMethod) ∧ Steaming(NextRecipeCookingMethod)
	6		Recipe(?r) ∧ Frying(?m) ∧ requiresCookingMethod(?r, ?m)	Frying(NextRecipeCookingMethod)
Entity Mapping	7	Appliance(?p) ∧ sosa:hosts(?p, ?s) ∧ sosa:Sensor(?s) ∧ sosa:observes(?s, ?ft) ∧ FoodType(?ft) ∧ madeLastObservation(?s, ?ob) ∧ sosa:Observation(?ob) ∧ sosa:hasSimpleResult(?ob, ?res)	Food(?f) ∧ hasDescription(?f, ?res)	sosa:hosts(?p, ?f)
	8	Appliance(?p) ∧ Oven(?p)	MILKCreateTool(InputProcedure) ∧ usesTool(InputProcedure, ?t) ∧ Oven(?t)	sameAs(?t, ?p) ∧ ShowNextRecipeStep(OutputProcedure)
	9	Appliance(?p) ∧ Refrigerator(?p)	MILKCreateTool(InputProcedure) ∧ usesTool(InputProcedure, ?t) ∧ Refrigerator(?t)	sameAs(?t, ?p) ∧ ShowNextRecipeStep(OutputProcedure)
	10	Appliance(?p) ∧ Stove(?p)	MILKCreateTool(InputProcedure) ∧ usesTool(InputProcedure, ?t) ∧ Stove(?t)	sameAs(?t, ?p) ∧ ShowNextRecipeStep(OutputProcedure)
	11	Tool(?p)	MILKCreateIng(InputProcedure) ∧ from(InputProcedure, ?p)	ShowNextRecipeStep(OutputProcedure)
Environment Monitoring	12	Tool(?pl)	MILKPut(InputProcedure) ∧ usesTool(InputProcedure, ?pl) ∧ usesFoodIngredient(InputProcedure, ?f)	willMoveTo(?f, ?pl)
	13	Tool(?p) ∧ Food(?f) ∧ sosa:hosts(?p, ?f)	MILKProcedure(InputProcedure) ∧ usesFoodIngredient(InputProcedure, ?f)	from(InputProcedure, ?p)

	14	Food(?x) ∧ Food(?y)	generatesFoodIngredient(?p, ?y) ∧ usesFoodIngredient(?p, ?x) ∧ MILKProcedure(?p)	willTransformTo(?x, ?y)
	15	Appliance(?p1) ∧ Tool(?p2) ∧ sosa:hosts(?p1, ?f) ∧ sosa:Sensor(?s) ∧ Weight(?w) ∧ sosa:hosts(?p1, ?s) ∧ sosa:observes(?s, ?w) ∧ madeLastObservation(?s, ?ob1) ∧ madeNewObservation(?s, ?ob2) ∧ sosa:hasSimpleResult(?ob1, ?v1) ∧ sosa:hasSimpleResult(?ob2, ?v2) ∧ swrlb:lessThan(?v2, ?v1)	willMoveTo(?f, ?p2)	ShowNextRecipeStep(OutputProcedure)
Appliance Automation	16	Appliance(?p) ∧ sosa:hosts(?p, ?h) ∧ sosa:Actuator(?h) ∧ Temperature(?temp) ∧ Timer(?timer) ∧ sosa:actsOnProperty(?h, ?temp) ∧sosa:actsOnProperty(?h, ?timer)	MILKCook(InputProcedure) ∧ usesTool(InputProcedure, ?t)	sameAs(?temp, OutputProperty1) sameAs(?timer, OutputProperty2) ∧ sameAs(?h, OutputActuator) ∧ sameAs(?p, OutputPlatform) ∧
	17	Appliance(?p) ∧ sosa:hosts(?p, ?h) ∧ sosa:Actuator(?h) ∧ Temperature(?temp) ∧ Timer(?timer) ∧ sosa:actsOnProperty(?h, ?temp) ∧ sosa:actsOnProperty(?h, ?timer) ∧ Cookware(?t)	MILKCook(InputProcedure) ∧ usesTool(InputProcedure, ?t)	sameAs(?temp, OutputProperty1) sameAs(?timer, OutputProperty2) ∧ sameAs(?h, OutputActuator) ∧ sameAs(?p, OutputPlatform)
	18	Heater(?ac) ∧ Appliance(?t) ∧ sosa:hosts(?t, ?ac) ∧ Temperature(?p1) ∧ sosa:actsOnProperty(?ac, ?p1)	MILKSet(InputProcedure) ∧ usesTool(InputProcedure, ?t)	sameAs(?t, OutputPlatform) ∧ sameAs(?ac, OutputActuator) ∧ sameAs(?p1, OutputProperty1)
	19	Appliance(?p) ∧ Content(?c) ∧ Display(?d) ∧ sosa:actsOnProperty(?d, ?c) ∧ sosa:hosts(?p, ?d)	MILKPut(InputProcedure) ∧ usesTool(InputProcedure, ?p)	sameAs(?p, OutputPlatform) ∧ sameAs(?d, OutputActuator) ∧ sameAs(?c, OutputProperty1)
	20	Appliance(?p) ∧ Content(?c) ∧ Display(?d) ∧ sosa:actsOnProperty(?d, ?c) ∧ sosa:hosts(?p, ?d)	MILKPut(InputProcedure) ∧ from(InputProcedure, ?p)	sameAs(?p, OutputPlatform) ∧ sameAs(?d, OutputActuator) ∧ sameAs(?c, OutputProperty1)

The table is listed in the format that *Appliance Condition* ∧ *Cooking Condition* → *Inference*;
$?x$ represents a variable that can be any individual in the ontology; ∧ represents conjunction of predicates

ACKNOWLEDGMENT

This work was sponsored in part by the NSF under CNS-1439717 with additional support from the NSF Center for Robots and Sensors for the Human well-Being (RoSe-HUB) and Tupperware Corp.

REFERENCES

[1] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer*, vol. 48, no. 1, pp. 28–35, 2015.

[2] K. Ashton *et al.*, "That 'Internet of Things' thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[4] G. Bell and J. Kaye, "Designing technology for domestic spaces: A kitchen manifesto," *Gastronomica*, vol. 2, no. 2, pp. 46–62, 2002.

[5] L. Terrenghi, O. Hilliges, and A. Butz, "Kitchen stories: sharing recipes with the living cookbook," *Personal and Ubiquitous Computing*, vol. 11, no. 5, pp. 409–414, 2007.

[6] M. Schneider, "The semantic cookbook: sharing cooking experiences in the smart kitchen," in *Proceedings of the 3rd International Conference on Intelligent Environments (IET)*, 2007, pp. 416–423.

[7] M. Ständer, A. Hadjakos, N. Lochschmidt, C. Klos, B. Renner, and M. Mühlhäuser, "A smart kitchen infrastructure," in *Proceedings of the 14th IEEE International Symposium on Multimedia (ISM)*, 2012, pp. 96–99.

[8] R. Blasco, Á. Marco, R. Casas, D. Cirujano, and R. Picking, "A smart kitchen for ambient assisted living," *Sensors*, vol. 14, no. 1, pp. 1629–1653, 2014.

[9] F. Gullà, S. Ceccacci, R. Menghi, and M. Germani, "An adaptive smart system to foster disabled and elderly people in kitchen-related task," in *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, 2016, pp. 1–4.

[10] A. Neumann, C. Elbrechter, N. Pfeiffer-Leßmann, R. Kõiva, B. Carlmeyer, S. Rüther, M. Schade, A. Üçkermann, S. Wachsmuth, and H. J. Ritter, "KogniChef: A cognitive cooking assistant," *KI - Kunstliche Intelligenz*, vol. 31, pp. 273–281, Aug. 2017.

[11] K. Achary, P. Auti, P. Khyadgi, and S. Korade, "A smart kitchen device using ultrasonic sensor for storage of food ingredient in mega kitchens," *International Reseach Journal of Engineering and Technology*, vol. 4, pp. 568–570, 2017.

[12] S. U. Shariff, M. Gurubasavanna, and C. Byrareddy, "IoT-based smart food storage monitoring and safety system," in *Proceedings of the 2018 International Conference on Computer Networks and Communication Technologies (ICCNCT)*, pp. 623–638.

[13] Y. B. Jinila, V. Rajalakshmi, L. M. Gladence, and V. M. Anu, "Food consumption monitoring and tracking in household using smart container," in *Proceedings of the Third International Conference on Computational Intelligence and Informatics*. Springer, Singapore, 2020, pp. 693–700.

[14] "The pantrychic smart storage system will change the way you look at storing and preparing recipes," *PantryChic*, Accessed on Nov. 29, 2021. [Online]. Available: https://www.pantrychic.com/

[15] "Keep tabs on your food like never before," *Ovie*, Accessed on Nov. 29, 2021. [Online]. Available: https://ovie.life/home

[16] "The world's most advanced smart cooking system," *Hestan Cue*, Accessed on Nov. 29, 2021. [Online]. Available: https://www.hestancue.com/

[17] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, pp. 2787–2805, 2010.

[18] P. Webster, V. Uren, and M. Ständer, "Shaken not stirred: mixing semantics into XPDL," in *CEUR Workshop Proceedings*, vol. 682, 2010, pp. 29–35.

[19] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, pp. 34–43, May 2001.

[20] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the internet of things: An overview from the inter-iot perspective," *Journal of Network and Computer Applications*, vol. 81, pp. 111–124, 2017.

[21] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "IoT-Lite: a lightweight semantic model for the Internet of Things and its use with dynamic semantics," *Personal and Ubiquitous Computing*, vol. 21, no. 3, pp. 475–487, 2017.

[22] R. Agarwal, D. G. Fernandez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sanchez, N. Georgantas, and V. Issarny, "Unified IoT ontology to enable interoperability and federation of testbeds," *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 70–75, Feb. 2017.

[23] G. Bajaj, R. Agarwal, P. Singh, N. Georgantas, and V. Issarny, "A study of existing ontologies in the IoT-domain," *arXiv*, July 2017. [Online]. Available: http://arxiv.org/abs/1707.00112

[24] T. G. Stavropoulos, D. Vrakas, D. Vlachava, and N. Bassiliades, "BOnSAI: a smart building ontology for ambient intelligence," in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, 2012, pp. 1–12.

[25] M. Waibel, M. Beetz, J. Civera, R. d'Andrea, J. Elfring, D. Galvez-Lopez, K. Häussermann, R. Janssen, J. Montiel, A. Perzylo *et al.*, "Roboearth," *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, 2011.

[26] R. M. Voyles and J. Bae, "Smart Tupperware: An example of bluetooth wireless sensor networks for human assistive mechatronic systems," in *Proceedings of the International Conference on Advanced Robotics*. Citeseer, 2007, pp. 449–454.

[27] R. Voyles, J. Bae, B. Smith, D. Kusuma, and L. Nguyen, "Smart Tupperware: Active containers for kitchen automation," in *Proceedings of the SICE Annual Conference*, 2008, pp. 3065–3069.

[28] S. Eom, R. M. Voyles, and D. Kusuma, "Embedding intelligence into smart tupperware brings internet of things home," *SPE Annual Technical Conference, Detroit, MI*, 2019.

[29] "FridgeSmart® Tupperware containers," *Tupperware*, Accessed on Nov. 29, 2021. [Online]. Available: https://www.tupperware.com/collections/fridgesmart/

[30] "Paragon: Make precision cooking easier with the Paragon smart cooking system," *FirstBuild*, Accessed on Nov. 29, 2021. [Online]. Available: https://support.firstbuild.com/hc/en-us/categories/203029728-Paragon

[31] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, Apr. 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[32] K. Janowicz, A. Haller, S. J. Cox, D. L. Phuoc, and M. Lefrançois, "Sosa: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics*, vol. 56, pp. 1–10, May 2019.

[33] D. Tasse and N. A. Smith, "SOUR CREAM: Toward semantic processing of recipes," *Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-LTI-08-005*, 2008.

[34] "A free, open-source ontology editor and framework for building intelligent systems," *Protégé*, Accessed on Nov. 29, 2021. [Online]. Available: https://protege.stanford.edu/

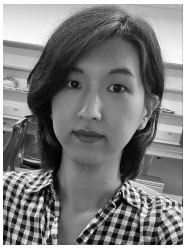
[35] Z. Zhai, J. F. M. Ortega, N. L. Martínez, and P. Castillejo, "A rule-based reasoner for underwater robots using OWL and SWRL," *Sensors*, vol. 18, no. 10, p. 3481, Oct. 2018.

[36] M. Horridge and S. Bechhofer, "The OWL API: A Java API for OWL ontologies," *Semantic Web*, vol. 2, no. 1, pp. 11–21, 2011.

[37] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, "HermiT: an OWL 2 reasoner," *Journal of Automated Reasoning*, vol. 53, no. 3, pp. 245–269, 2014.

[38] "Allrecipes: Food, friends, and recipe inspiration," *AllRecipes.com*, Accessed on Nov. 29, 2021. [Online]. Available: https://www.allrecipes.com/

[39] "Recipes, cookbooks, 1 million recipe database," *Cookbooks.com*, Accessed on Nov. 29, 2021. [Online]. Available: https://www.cookbooks.com/



Sangjun Eom (Graduate Student Member, IEEE) received the B.S., in 2017, and M.S. degrees in Electrical Engineering Technology from Purdue University, West Lafayette, IN, USA, in 2020. She is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at Duke University, NC, USA. Her current research interests include Augmented Reality and Internet of Things.

Haozhe Zhou is currently pursuing the B.S. degree in computer science at Purdue University, West Lafayette, IN, USA. His current research interests include ubiquitous computing, machine learning, human-computer interaction, computer security and privacy.

Upinder Kaur (Graduate Student Member, IEEE) is currently a Ph.D. student at Polytechnic Institute, Purdue University. Her research interests are in reinforcement learning, computer vision, learning from demonstration, and Cyber-Physical Systems with robotic edges.

Richard M. Voyles (Fellow, IEEE) is Professor of Engineering Technology and University Faculty Scholar at Purdue University. Dr. Voyles is the founder of the Collaborative Robotics Lab, director of the Purdue Robotics Accelerator, and site director of the NSF Center for Robotics and Sensors for Human Well-Being (RoSe-HUB). He holds a B.S. in Electrical Engineering from Purdue University, in 1983, M.S. in Mechanical Engineering at Stanford University, in 1989, and Ph.D. in Robotics at Carnegie Mellon University, in 1997. Dr. Voyles' research interests include miniature, constrained robots; mobile manipulation; multi-robot coordination; programming by human demonstration; and haptic sensors and actuators.

David Kusuma is a Vice President of Research and Product Innovation at Tupperware Brands Corporation. Dr. Kusuma received a B.F.A from Carnegie Mellon University, Pittsburgh, PA, USA, in 1986, M.S. from Purdue University, West Lafayette, IN, USA, in 2002, and Ph.D. from Cranfield University, UK, in .