

SPARCS: Structuring Physically Assistive Robotics for Caregiving with Stakeholders-in-the-loop

Rishabh Madan^{*†}, Rajat Kumar Jenamani^{*†}, Vy Thuy Nguyen[†], Ahmed Moustafa[†], Xuefeng Hu[†], Katherine Dimitropoulou[‡], Tapomayukh Bhattacharjee[†]

Abstract—Existing work in physical robot caregiving is limited in its ability to provide long-term assistance. This is majorly due to (i) lack of well-defined problems, (ii) diversity of tasks, and (iii) limited access to stakeholders from the caregiving community. We propose Structuring Physically Assistive Robotics for Caregiving with Stakeholders-in-the-loop (SPARCS) to address these challenges. SPARCS is a framework for physical robot caregiving comprising (i) *Building Blocks*, models that define physical robot caregiving scenarios, (ii) *Structured Workflows*, hierarchical workflows that enable us to answer the *Whats* and *Hows* of physical robot caregiving, and (iii) *SPARCS-box*, a web-based platform to facilitate dialogue between all stakeholders. We collect clinical data for six care recipients with varying disabilities and demonstrate the use of SPARCS in designing well-defined caregiving scenarios and identifying their care requirements. All the data and workflows are available on *SPARCS-box*. We demonstrate the utility of SPARCS in building a robot-assisted feeding system for one of the care recipients. We also perform experiments to show the adaptability of this system to different caregiving scenarios. Finally, we identify open challenges in physical robot caregiving by consulting care recipients and caregivers. Supplementary material can be found at emprise.cs.cornell.edu/sparcs.

I. CAREGIVING IS MULTIFACETED AND CONTEXTUAL

Morgan (he/him) is a 58-year-old who sustained a brain-stroke at 40 that made him quadriplegic and mute. He receives assistance from his partner and primary caregiver Riley (she/her), and his professional caregiver Spencer (they/them)¹. Morgan’s caregiving is complex and personalized. It is extremely challenging for any arbitrary person without proper training to assist him. His assistance is contextual on his functional abilities and behavior, his caregiver, and his environment:

1. *Care Recipient Context*: When being assisted with feeding, Morgan turns to his caregiver to show his intent to have a bite. The mobility limitations in his tongue necessitate solid food to be placed around his lower left molar for chewing.
2. *Caregiver Context*: Morgan is tall. Dressing him involves lifting his limbs. While Spencer is tall and muscular and does not face difficulties with this task, Riley performs it differently because she is short and lean.
3. *Environment Context*: Morgan lives in a house with a small bathroom. Bathing Morgan is especially challenging due to

^{*}Equal contribution

[†]Rishabh Madan, Rajat Kumar Jenamani, Vy Nguyen, Ahmed Moustafa, Xuefeng Hu, Tapomayukh Bhattacharjee are with the Department of Computer Science, Cornell University, Ithaca, NY, USA, {rm773, rj277, vtn39, aem278, xh244, tb557}@cornell.edu

[‡]Katherine Dimitropoulou is with Columbia University, New York City, NY, USA, kd2524@cumc.columbia.edu

¹These are real people whom we interviewed as part of a user study. We use fictitious names/pronouns for this paper.



Fig. 1: SPARCS enables roboticists to translate human caregiving requirements to guidelines for physical robot caregiving.

his occasional spasms. These spasms lead to involuntary movements, which can hurt him if his limbs hit the bathroom walls. This requires his caregivers to be extremely vigilant.

Many of Morgan’s activities of daily living (ADLs) easily take more than an hour and pose a challenge to all other family routines. Caregiving for Morgan is very taxing for Riley and leaves her with barely any time for her own needs. Morgan is aware of the situation and wishes to do some of these activities by himself, not only to feel more independent but also to reduce Riley’s burden.

II. INTRODUCTION

Morgan is among many others worldwide who require assistance with ADLs. Nearly 27% of people living in the United States have a disability, and close to 24 million people aged 18 years or older need assistance with ADLs [1]. Similar to Riley and Spencer, there are several family and professional caregivers who are overburdened and experience mental stress and declining quality of life [2]–[4]. Robots have the potential to provide assistance with ADLs [5] and empower people with disabilities by enhancing their independence [6], while also reducing caregiver burden.

Existing work in physical robot caregiving is limited in its ability to consistently provide in-situ assistance to care recipients for the entire duration of an ADL. This is due to a myriad of factors such as diversity of tasks, the need for personalized assistance, and limited access to other stakeholders (care recipients, their caregivers, and occupational therapists) who have the required domain knowledge. There exists variability in problem definitions, which points to an inherent lack of common grounding in problem setup. Without a common underlying framework to systematically reason about physical robot caregiving, it is very challenging to transfer insights from one specific scenario to another.

We propose Structuring Physically Assistive Robotics for Caregiving with Stakeholders-in-the-loop, or **SPARCS** to address these challenges. This framework comprises **Building Blocks** (Sec. III), **Structured Workflows** (Sec. IV), and **SPARCS-box** (Sec. V). Inspired by the different contexts

in human caregiving, we define four *Building Blocks*, the *user*, i.e., the care recipient, their *human caregiver*, their *environment*, and the *robot*. We build upon these blocks to form *Structured Workflows*, hierarchical workflows that enable us to answer the *whats* and *hows* of physical robot caregiving. Building these workflows requires communication between roboticists and other stakeholders who have expertise in caregiving. *SPARCS-box* is a web-based platform that facilitates this dialogue.

SPARCS is a framework for building physical caregiving robots. We perform experiments that highlight the importance of *Building Blocks*, particularly user models, to build effective and personalized systems. We show SPARCS in practice by using it to (i) identify care requirements, and (ii) utilize these requirements to build a physical caregiving robot. We consider six care recipients with varying disabilities and collect data on the *Building Blocks* for ADLs that they require assistance with. We perform a user study with expert occupational therapists to build the *whats* of *Structured Workflows* for all the identified caregiving scenarios. This can help us translate the caregiving needs of the six care recipients into well-defined research problems for roboticists. All the data and workflows are publicly available on *SPARCS-box* [7] and can be used by roboticists to reason about the *hows* for robot caregiving. We illustrate the process of generating physical robot caregiving solutions for one of the scenarios - the activity of robot-assisted feeding for one of the six care recipients. Additionally, we demonstrate how this feeding system adapts to changes in the different building blocks. Finally, we conduct a study with care recipients and caregivers, asking them questions on general ADL assistance. We use SPARCS to examine their caregiving needs and share open challenges for the robotics community.

We summarize our contributions as follows:

- We propose SPARCS, a framework for physical robot caregiving. SPARCS comprises (i) *Building Blocks*, models that define physical robot caregiving scenarios, (ii) *Structured Workflows*, hierarchical workflows that enable us to answer the *whats* and *hows* of physical robot caregiving, and (iii) *SPARCS-box*, a web-based platform that facilitates communication between all stakeholders.
- We highlight the importance of *Building Blocks*, particularly user models, to build effective and personalized physical caregiving robots.
- We collect real-world data and create *Building Blocks* for six care recipients with varying disabilities. We conduct a user study with occupational therapists and use *Structured Workflows* to identify care requirements for these care recipients.
- We demonstrate the use of SPARCS for building a robot-assisted feeding system for one of the six care recipients. Additionally, we exhibit the adaptive nature of this system through reuse and transfer of its *Structured Workflows* across different *Building Blocks*.
- We identify open challenges in physical robot caregiving for roboticists through another user study with care recipients and caregivers.

III. BUILDING BLOCKS OF SPARCS

Designing a caregiving robot involves reasoning about four *Building Blocks* – the **User**, their **Human Caregiver**, their **Environment**, and the **Robot**. We define these blocks using functionality and behavioral user models, corresponding caregiver models, an environment model, and a robot model.

A. User Functionality Model

The *User Functionality Model* M_{UF} represents a care recipient's physical and cognitive functioning abilities.

Physical functioning should capture body shape and structure, muscle function, joint limits, and involuntary movements. Cognitive functioning should model mental capabilities such as attention, memory, and decision-making abilities. The *Human Caregiver Functionality Model* M_{CF} follows a similar definition, but for the caregiver.

Body shape and structure in M_{UF} can be represented through articulated human-shape models such as SMPL-X [8]. Muscle function can be incorporated using high-fidelity musculoskeletal models [9]–[11]. Joint limits are pose-dependent and affected by the mobility limitations of the user. They can be approximated by existing learning-based approaches [12,13] trained on user movement data. Involuntary movements have been previously simulated using parameterized offset functions [14]. All of the above models for physical functioning can be adapted to user measurements such as body dimensions, range of motion (ROM) [15], and muscle strength [16]. Cognitive functioning of the user can be characterized using measures such as MMSE [17] or SLUMS [18]. The International Classification of Functioning, Disability and Health [19] framework defines body structures and functions, and is employed by professionals from various fields including health, rehabilitation, and community care. It can be used to model M_{UF} more extensively.

B. User Behavioral Model

The *User Behavioral Model* M_{UB} represents a care recipient's intent and preferences, which is necessary for providing personalized care [20]. The *Human Caregiver Behavioral Model* M_{CB} follows a similar definition, but for the caregiver.

Intent can be identified through explicit communication or implicit inference. Explicit communication can be established through an interface, such as speech or a GUI [21]. Implicit inference can be made using Bayesian models [22,23] or modeling in latent space [24]. Preferences can be global or task-environment specific [25]. For example, a user may always prefer the robot to be slow-moving regardless of the scenario, whereas their preference for the level of autonomy may be task dependent. Preferences can be user-specified [26] or modeled using data-driven methods [27].

C. Environment Model

The *Environment Model* M_E represents physical and social information about the user's environment. Physical information constitutes the surrounding scene and the objects (including assistive devices) present in it. Social information records the environment's social context. For example, it

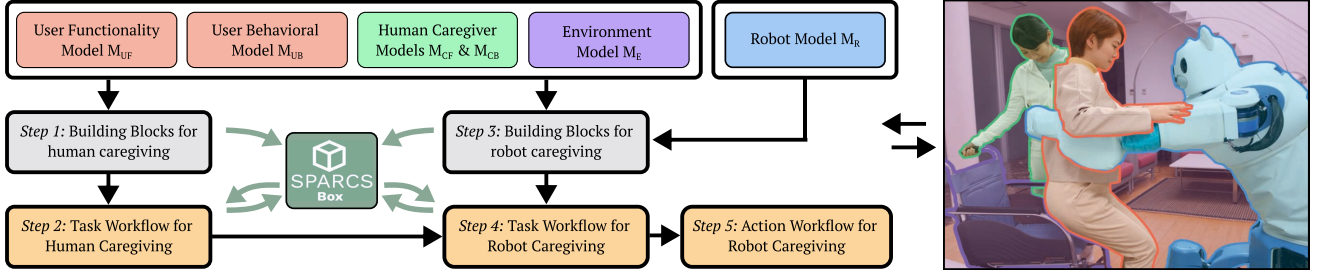


Fig. 2: Using SPARCS involves the following steps: 1. Instantiate the *Building Blocks* for human caregiving, 2. Communicate with stakeholders using *SPARCS-box* to create the *Task Workflow for Human Caregiving*, 3. Instantiate the *Building Blocks* for robot caregiving, 4. Propose the *Task Workflow for Robot Caregiving* and incorporate feedback from stakeholders using *SPARCS-box*, and 5. Implement the *Action Workflow for Robot Caregiving*. Image shows ROBEAR from RIKEN.

could be an intrapersonal setting involving only the care-recipient and their caregiver, or an interpersonal setting with family members, or a community setting [28].

Physical information is necessary for most robotic applications. Environment scenes can be represented using topological maps and semantic point clouds, whereas objects can be encoded using Unified Robot Description Format (URDF) files [29]. Physical information can also be represented using more detailed data structures such as 3D scene graphs [30]. Social information can be captured through theory of mind approaches [31] or data-driven methods [32].

D. Robot Model

The *Robot Model* M_R represents the hardware specifications, kinematics and dynamics, visual and collision model, and onboard sensing capabilities of the robot. It also provides access to the raw sensor data. It is important to explicitly model the robot as it affects ADL assistance [28].

URDFs are commonly used to represent many of the aforementioned attributes. M_R can consist of these along with other configuration files that contain information on sensors and their metadata.

For the experiments in this paper, we initialize each *Building Block* $M: K \rightarrow V$ as a dictionary where K and V are mutable sets of keys and values, respectively. K contains keywords corresponding to the attributes of the *Building Block*. V contains instantiations of these attributes. For example, in case of M_{UF} , K may contain keywords like “Active ROM Neck Flexion” or “Passive ROM Neck Extension” with corresponding values stored in V .

IV. STRUCTURED WORKFLOWS: WHATS AND HOWS OF ROBOT CAREGIVING

Given the *Building Blocks*, how do we enable the robot to provide assistance for the entire duration of an ADL? This requires understanding the *whats* that the robot must address in the caregiving scenario. We call these *whats*, i.e., the set of tasks, the *Task Workflow for Robot Caregiving*.

While creating the *Task Workflow for Robot Caregiving*, roboticists must consult other stakeholders who have expertise in caregiving. However, many of these stakeholders have no experience with robotics. They cannot propose robot caregiving workflows directly. Therefore, we advocate first getting their insights on the corresponding human-human

caregiving scenario. We call this the *Task Workflow for Human Caregiving*. Roboticists can then use this information to propose an initial *Task Workflow for Robot Caregiving*, and incorporate feedback from other stakeholders on it.

Once the *Task Workflow for Robot Caregiving* is finalized, the focus can shift to answering *how* a robot can perform the constituent tasks. We call this implementation workflow the *Action Workflow for Robot Caregiving*. The two *Task Workflows* and this *Action Workflow* are jointly referred to as *Structured Workflows*.

What would be a good representation for *Structured Workflows*? The representation must allow enough detailing to capture task-related considerations. For example, when feeding Morgan, his caregiver applies downward forces on Morgan’s tongue to avoid gag reflex. At the same time, the representation should be hierarchical such that these details can be gradually uncovered as we move through various layers of abstraction. Hierarchy will also promote the reusability of similar subroutines across different caregiving scenarios. For example, the subroutine of lifting a user’s leg while dressing with pants can be reused for lifting their leg during a sponge bath. The representation must also support the specification of concurrent tasks like wiping off water from the user’s eyes while applying shampoo on their head. Finally, the representation must be easy to understand and update for facilitating valuable discussions among all stakeholders.

Hierarchical State Machines (HSMs) [33] satisfy all of the above requirements. They offer an intuitive visual formalism for complex systems with various levels of abstraction. We propose to use HSMs as the representation for both *Task* and *Action Workflows*. Depending on the layer of abstraction, states of this machine define the routines or how they are carried out at the corresponding level of detail. Each state has pre-conditions and post-conditions that govern transitions in and out of it.

We define the abstractions for the two *Task Workflows* in a top-down manner:

$$\text{Activity} \rightarrow \text{Composite Task} \rightarrow \text{Task}$$

An *Activity* is an instantiation of a given ADL. It maps to a set of *Composite Tasks* depending on the *User Functionality Model* and the initial state of the *Environment Model*. *Composite Tasks* further split into *Tasks* conditioned on all the *Building Blocks*.

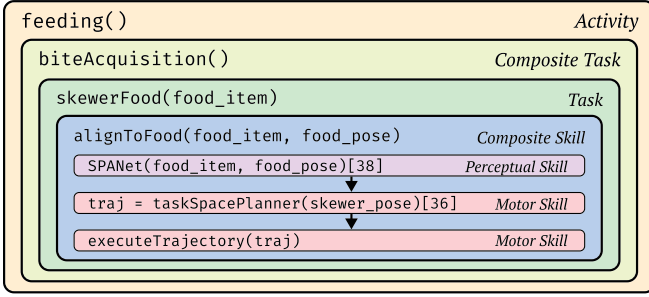


Fig. 3: Snippet of a *Structured Workflow* for robot-assisted feeding showing all the layers of abstraction.

In case of *Task Workflow for Robot Caregiving*, we map the constituent *Tasks* to their corresponding *hows*, i.e., the *Action Workflow for Robot Caregiving*. We define the abstractions for *Action Workflows* as:

Composite Skill \rightarrow *Motor/Perceptual Skill*

Each *Task* maps to a set of *Composite Skills*. *Composite Skill* further comprises *Motor* and *Perceptual Skills*. *Motor Skills* represent the robot’s ability to reason about its physical movements. *Perceptual Skills* represent the robot’s ability to interpret sensor data. Each layer of abstraction in *Structured Workflows* is well-scoped with respect to its parent as detailed on our website [34]. Figure 3 shows a snippet of a *Structured Workflow* for robot-assisted feeding.

V. SPARCS-BOX: BRIDGING THE GAP BETWEEN ROBOTICISTS AND CAREGIVING COMMUNITY

Building *Structured Workflows* requires roboticists to consult other stakeholders who have the required domain knowledge in caregiving. However, this is currently challenging due to limited access to stakeholders. To alleviate this problem, we propose *SPARCS-box* [7], a web-based platform that facilitates discussion between all stakeholders.

SPARCS-box allows stakeholders to sign up and propose caregiving scenarios of their choice. A caregiving scenario can be specified by detailing the *Activity* and its *Building Blocks*. Stakeholders can collaborate on the platform to design its *Task Workflow for Human Caregiving*. Roboticists can then use this information to propose the corresponding *Task Workflow for Robot Caregiving*. They can get feedback on it from other stakeholders on the platform. Once this workflow is finalized, roboticists can work on well-defined robot caregiving problems. They can pick *Tasks* from this workflow and build *Skills* that contribute to the *Action Workflow for Robot Caregiving*. We provide more details on all the functionalities of *SPARCS-box* on our website [34].

VI. IMPORTANCE OF USER MODELING IN SPARCS

Physical robot caregiving requires instantiating detailed *User Functionality Model* M_{UF} and *User Behavioral Model* M_{UB} . These models help in building more effective and personalized systems. To illustrate this, we consider *bite transfer* and *bite sequencing* for a robot-assisted feeding activity with a wheelchair-mounted robot arm. Bite transfer involves transferring a food item from a utensil into the mouth of a care recipient. Bite sequencing involves deciding the next food item to feed a care recipient. We demonstrate

how algorithms that leverage user-informed a) M_{UF} for bite transfer and b) M_{UB} for bite sequencing perform better than baselines.

A. Leveraging M_{UF} for Bite Transfer

To show the effect of M_{UF} for bite transfer, we select the user Natalia (Table I) for our experiments.

Experiment setup: Our bite transfer policy transfers a food item from its initial fixed position facing Natalia to near her mouth. Her head pose is denoted by h_{user} . We use active ROM data for her neck to instantiate the manifold $H_{M_{UF}}$ of attainable head poses. We denote the set of all head poses attainable by an individual with full neck mobility by H_{all} . We consider the test set spanning all possible $h_{user} \in H_{M_{UF}}$.

Methods: We compare three bite transfer control policies: (i) Fixed [35] which executes a trajectory to an assumed fixed head pose $h_{fixed} \in H_{M_{UF}}$, (ii) Baseline which uses H_{all} , and (iii) M_{UF} -informed which uses $H_{M_{UF}}$.

M_{UF} -informed begins with sampling a set H_{cand} comprising candidate head poses from M_{UF} where the bite transfer can happen. H_{cand} also includes h_{user} and h_{fixed} . These poses are ordered according to relative angle from h_{user} . The algorithm sequentially iterates over each sample in H_{cand} . For each head pose sample $h_{cand} \in H_{cand}$, the corresponding end effector goal pose x_{goal} is found. Natalia aligns her mouth opposite to the fork while taking the food item. Thus, x_{goal} is assumed to be at a fixed transform to h_{cand} . We then use task-space-region planner [36] to find a collision-free trajectory from the initial configuration of the robot to x_{goal} . If the trajectory is successfully found, it is returned. If no trajectory is found even after iterating over all the samples, the algorithm terminates.

Baseline follows a similar approach. However, as it does not have access to $H_{M_{UF}}$ or h_{fixed} , it instead constructs H_{cand} using h_{user} and head poses sampled from H_{all} .

Metrics: Performance of the above methods is averaged over ten seeds for the test set of sampled h_{user} and evaluated on two metrics. (1) *Success rate*, which is the percentage of test samples where the bite transfer control policy can generate a feasible trajectory. A trajectory is said to be feasible if it is collision-free and reaches a pose that results in successful bite transfer. (2) *Relative angle* by which a user has to move their neck to take the bite off the fork.

Results: Among the compared policies, M_{UF} -informed can always successfully find a feasible control policy and requires the smallest neck movement (Fig. 4). Our experiments illustrate that access to user-informed M_{UF} leads to more efficient physical robot caregiving.

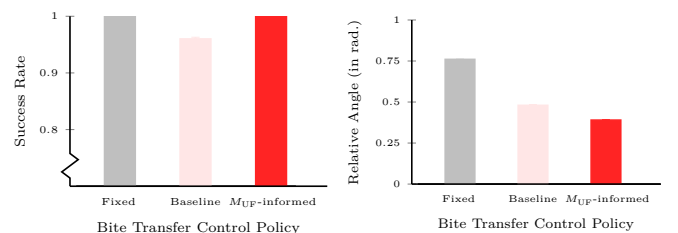


Fig. 4: Fixed and M_{UF} -informed policies have a high bite transfer success rate, while the latter also results in minimal neck movement.

B. Leveraging M_{UB} for Bite Sequencing

We show the effect of modeling M_{UB} for bite sequencing on user satisfaction. Among a set of candidate models for capturing M_{UB} , we perform a user study to identify the model that has the highest overall user satisfaction.

Experiment setup: We consider a meal consisting of three bites each of four unique food items – banana, kiwi, grape, and carrot. Assuming the user eats one item at a time and consumes all the food items, the bite sequence for a meal is an ordered sequence of these food items. We perform a user study with 14 participants. For each participant, we initially collect two data points. First, an affinity score $\in [1, 5]$ for each food item that implies how much they like it. Second, their high-level eating preference among (a) *I would save my favorite food for the last*, (b) *I would eat my favorite food first*, and (c) *I prefer to mix and match*. We then ask the participants to record their bite sequences for six meals. Using this data, the goal is to learn a preference model as part of M_{UB} and use it for predicting the bite sequence during the seventh meal.

Methods: Our approach comprises (i) generating simulated bite sequences for a user using their affinity score and high-level eating preference, (ii) training a hidden Markov Model (HMM) with discrete emissions using the simulated data, and (iii) updating the model using the six bite sequences obtained from the user. We use HMM-simulated (HS) to refer to the model obtained after (ii) and HMM-online (HO) for the final updated model after (iii). We compare with a baseline that randomly selects food items.

Metrics: Performance of the above methods is evaluated on two metrics. (1) *Prediction accuracy* during the seventh meal. This is calculated by presenting the predictions from all the methods as choices to the participant and comparing the predictions with the item selected by the participant. (2) For each method, we generate a bite sequence assuming the participant always chooses the item predicted by that method, and ask them to provide a *satisfaction rating* $\in [1, 5]$ for the generated sequence.

Results: We perform Kruskal-Wallis H-tests and Tukey HSD post-hoc tests and observe that HO significantly outperforms other methods in both accuracy and satisfaction rating (Fig. 5). This user study demonstrates that modeling the user preferences for bite sequencing and adapting it to user data leads to improved satisfaction. Refer to our website [34] for more details on the user study.

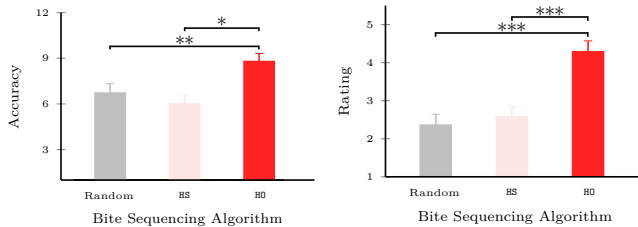


Fig. 5: HMM-online HO outperforms other methods in both accuracy and satisfaction rating. *, **, *** denote statistically significant differences with $p0.05$, $p0.005$, $p0.0005$ respectively.

VII. SPARCS IN PRACTICE: IDENTIFYING CARE REQUIREMENTS AND BUILDING A CAREGIVING ROBOT

SPARCS is a framework for building physical caregiving robots. Using SPARCS involves the following steps (Fig. 2):

A. Identifying care requirements:

1. Instantiate *Building Blocks* for human caregiving.
2. Communicate with stakeholders using *SPARCS-box* to create *Task Workflow for Human Caregiving*.

B. Building a physical caregiving robot:

3. Instantiate *Building Blocks* for robot caregiving.
4. Propose *Task Workflow for Robot Caregiving* and incorporate feedback from stakeholders using *SPARCS-box*.
5. Implement *Action Workflow for Robot Caregiving*.

We demonstrate each of these steps in this section. For steps 1 and 2, we consider six care recipients with varying functional abilities (Table I). We instantiate various caregiving scenarios for each of these care recipients and identify their care requirements (Sec. VII-A). For steps 3, 4, and 5, we consider one of the above identified scenarios – robot-assisted feeding of bite-sized food items to Natalia while she is watching television (Sec. VII-B).

A. Identifying Care Requirements for Six Care Recipients

We consider six care recipients with different functional abilities to capture variability in care requirements (Table I).

Step 1. Instantiate Building Blocks for Human Caregiving:

For each of the six care recipients, we instantiate corresponding *User Functionality Model* M_{UF} , *User Behavioral Model* M_{UB} , *Caregiver Models* (M_{CF} and M_{CB}), and *Environment Model* M_E . We collect clinical data for information on M_{UF} (Fig. 6) – body dimensions, weight, active and passive ROM [15] and manual muscle testing [16] – along with textual descriptions, and videos of non-disabled medical professionals simulating the functional abilities of these care recipients. For each care recipient, we look at ADLs – feeding, dressing, bathing, and transferring. Among these ADLs, we identify scenarios they require assistance with based on their functional abilities. We identify a total of 19 caregiving scenarios and provide the corresponding *Building Blocks*.

TABLE I: Information for the six care-recipients. We use initials of Feeding, Dressing, Bathing, and Transferring to denote their need for assistance with these ADLs.

Identifier	Cause of Disability	Needs Assistance
Morgan (he/him)	Brainstem Stroke	F, D, B, T
Jose (they/them)	Spinal Cord Injury (C1-C3)	F, D, B, T
Natalia (she/her)	Spinal Cord Injury (C4-C5)	F, D, B, T
Daniel (he/him)	Spinal Cord Injury (C6-C7)	D, T
Kim (she/her)	Cerebral Palsy	D, B, T
Karan (he/him)	Left-side Hemiplegia	D, T

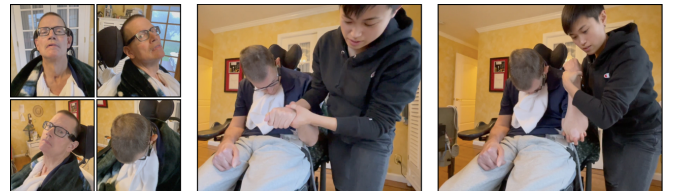


Fig. 6: Data Collection for Morgan's M_{UF} , left to right: Active Range of Motion, Passive Range of Motion, Manual Muscle Testing

Step 2. Create Task Workflow for Human Caregiving:

One should use SPARCS-box to create *Task Workflow for Human Caregiving*. For this paper, we conduct a user study interviewing 9 occupational therapists (2 male; 7 female), between the ages of 27 and 51. Through this study, we record *Task Workflows* for the 19 caregiving scenarios.

The *Building Blocks* and above workflows for all the identified caregiving scenarios are publicly available on SPARCS-box. This information can be leveraged by roboticists to build robots that can address the identified care requirements.

B. Building a Physical Caregiving Robot

In this section, we focus on the scenario of feeding Natalia while she is watching television.

Step 3. Instantiate Building Blocks for Robot Caregiving:

We instantiate the *Building Blocks* for the above scenario from the data collected in *Step 1* and include our *Robot Model* in it.

- *Environment Model M_E* : Natalia is sitting at the dining table on a wheelchair with a robot arm mounted beside its right armrest. The dining table is in her living room in-front of the television. There is a plate in front of her, consisting of solid bite-sized food items. This scene is captured using a pointcloud obtained from an RGBD camera. Objects in this scene are represented using URDFs [29]. The shape and pose of Natalia's head is represented using FLAME [37].
- *User Functionality Model M_{UF}* : Natalia has complete paralysis in all limbs. She has partial mobility in her neck. We use the data collected in *Step 1* to represent her M_{UF} .
- *User Behavioral Model M_{UB}* : Natalia prefers the robot to autonomously take decisions without any input from her, and shows her intent of taking a bite by opening her mouth. She tends to focus on the television while eating and thus favors making minimal neck movements to transfer the food from the fork to her mouth. Among the set of food items, she has a preferred order for eating the food items. She expects the robot to learn this ordering. We use HMM-online (Sec. VI-B) to model this preference.
- *Robot Model M_R* : We use the Kinova Gen3 6 degrees-of-freedom (DoF) robot arm with a Robotiq 2F-85 gripper, holding a custom fork fitted with an ATI Force/Torque sensor.

Step 4. Create Task Workflow for Robot Caregiving:

We build the *Task Workflow for Robot Caregiving* using the *Task Workflow for Human Caregiving* identified in *Step 2*. This *Activity* can be broken down into two *Composite Tasks*: (i) *Bite Acquisition*: acquiring a food item from the plate, and (ii) *Bite Transfer*: transferring this food item into the mouth of the care recipient. Natalia's M_{UB} specifies that she prefers the caregiver to feed her without any input. Thus, *Bite Acquisition* comprises the *Task* of the robot autonomously moving above the plate and then skewering the required food item. According to Natalia's M_{UF} , she has partial mobility in her neck. She can eat the food item off the fork if the robot brings it to a region within her functional abilities. *Bite Transfer* begins with the robot moving to a fixed pose in front of Natalia such that her head is visible to its camera.

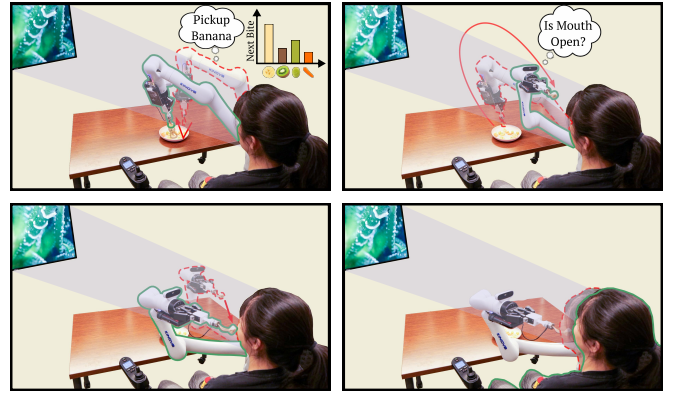


Fig. 7: Robot assisted-feeding for an individual simulating Natalia while she watches TV. *Top left*: Robot performs *Bite Acquisition* using SPANet [38]. *Top right*: It enters *Bite Transfer* and waits for the user to open mouth. *Bottom left*: It moves close to Natalia's mouth. *Bottom right*: Natalia leans forward and takes the bite.

According to her M_{UB} , the robot waits for her to open her mouth. Once it is open, the robot brings the food item near her mouth, ensuring minimal neck movement for transfer. Natalia then leans towards the food item and takes it off the fork.

Step 5. Implement Action Workflow for Robot Caregiving:

We demonstrate the *Action Workflow for Robot Caregiving* with an individual simulating Natalia. For *Bite Acquisition*, we deploy SPANet [38] for selecting how to skewer a food item from a set of candidate skewering actions. For *Bite Transfer*, we deploy the M_{UF} -informed policy (described in Sec. VI-A). The demonstration of our robot feeding an individual simulating Natalia can be seen in Fig. 7 and found on our website [34].

VIII. ADAPTING ROBOT-ASSISTED FEEDING TO DIFFERENT CAREGIVING SCENARIOS

In this section, we perform experiments to exhibit the adaptability of our system. We show how robot caregiving scenarios instantiated using SPARCS allow us to reuse and transfer insights across differing users, environments, and robot models.

A. User Functionality and Behavioral Models.

We adapt the *Structured Workflows* for Natalia to Jose (Table I) while keeping all other Building Blocks fixed:

- New M_{UF} : Jose has severe neck mobility limitations with partial neck rotation ROM. Unlike Natalia, they require their caregiver to place a food item inside their mouth cavity for *Bite Transfer*.
- New M_{UB} : Unlike Natalia, Jose shows their intent for wanting a bite by turning towards the robot, and expresses their consent for the food item to be placed inside their mouth by opening it. They prefer the food item to be placed one-third inside their mouth cavity. Similar to Natalia, they expect the robot to learn their preferred bite sequence.

In the *Structured Workflows* for this scenario, while *Bite Acquisition* remains similar, the *Task* breakdown for *Bite Transfer* changes. *Bite Transfer* begins with the robot moving to a fixed pose in-front of Jose such that their head is visible

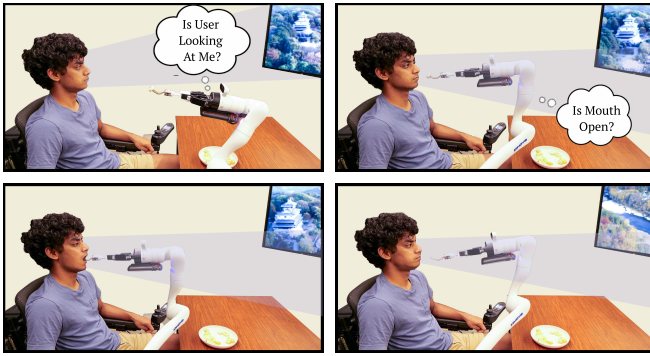


Fig. 8: Robot-assisted feeding for an individual simulating Jose. *Top left*: Jose watches television. *Top right*: They show the intent of taking a bite by looking at the robot. *Bottom left*: Robot performs an inside-the-mouth transfer due to Jose’s mobility limitations. *Bottom right*: Robot retracts leaving food item inside the mouth.

to its camera. In accordance with their M_{UB} , the robot waits for them to turn towards the robot and then moves the food item to in-front of their mouth. Once Jose’s mouth is open, the robot moves the food item inside their mouth cavity to successfully transfer the bite. The demonstration of our robot feeding an individual simulating Jose can be seen in Fig. 8 and found on our website [34].

B. Robot Model.

We consider a different *Robot Model* M_R while keeping the same user and environment models:

- New M_R : We use the Kinova Gen3 7-DoF robot arm with a Robotiq 2F-85 gripper, holding a custom fork fitted with an ATI Force/Torque sensor.

As the robot has a similar morphology to the one considered in Sec. VII-B, the *Structured Workflows* remain the same. However, the change in its dimensions and DoF affect its performance in *Bite Transfer* as shown in Table II. We observe that the new M_R performs better than the previous model which could be an effect of having better manipulability.

TABLE II: M_R comparison (averaged over three seeds)

Robot Arm	Success Rate	Relative Angle (in rad.)
Kinova Gen3 6-DoF	1.0	0.3996 ± 0.0018
Kinova Gen3 7-DoF	1.0	0.3496 ± 0.0008

C. Environment Model.

We consider a different *Environment Model* M_E while keeping the same user and robot models:

- New M_E : Natalia is sitting at her dining table on a wheelchair with a robot arm mounted beside its right armrest. There is a plate in-front of her, consisting of solid bite-sized food items. She is dining along with her friends in a social setting. While dining in this social setting, Natalia prefers the robot to bring the food item to a fixed position on her right side for *Bite Transfer*. This is to avoid any form of social distraction due to the robot motion. The demonstration of our robot feeding an individual simulating Natalia in the new M_E can be seen in Fig. 9 and found on our website [34].⁸

IX. OPEN CHALLENGES

As more roboticists get involved in physical robot caregiving, it is important to identify open problems that one can work on. Therefore, we reached out to care recipients and

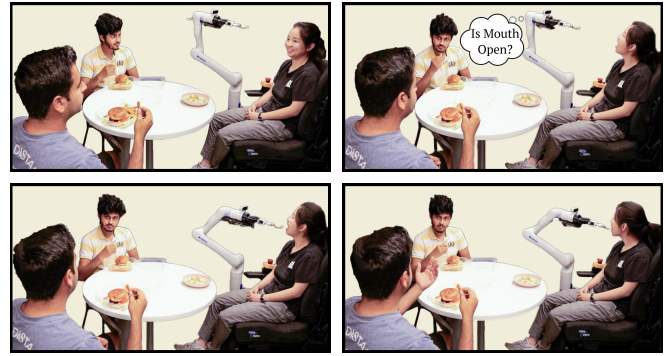


Fig. 9: Robot-assisted feeding for an individual simulating Natalia in a social dining scenario. *Top left*: Natalia having a conversation. *Top right*: In this M_E , Natalia faces the robot and opens her mouth. *Bottom left*: Robot moves close to her mouth. *Bottom right*: Natalia leans forward and takes the bite.

caregivers to better understand the critical factors that affect caregiving for ADLs. We used SPARCS to create caregiving scenarios around them and obtained recommendations on ADL assistance. We conducted a user study with 8 care recipients (6 male; 2 female) and 2 caregivers (both female), between the ages of 21 and 60. All data collection and user studies in this paper were approved by the Cornell University Institutional Review Board. Here we list some key open challenges:

Feeding. While there has been a lot of work on robot-assisted feeding, many challenges still need to be addressed for real-world applicability. Through our user study, we identify four unique challenges in feeding: (i) determining the level of autonomy because, it is not only personalized to a user but also to different tasks within feeding, (ii) acquiring a variety of food items with different physical characteristics across shape, size, texture, compliance, etc., (iii) designing the embodiment such that it can potentially perform other tasks while still being well suited for feeding, and (iv) determining when to switch between human supervision and autonomous control with our users preferring to switch to human supervision at the tail-end of a task (e.g. precisely aligning to a food item for *Bite Acquisition*) for finer control.

Dressing and Bathing. Shared autonomy methods purely based on joystick-based control have been considered viable for controlling caregiving robots. However, they may not be feasible for many tasks within dressing and bathing. For example, when dressing a user with a t-shirt, the user may not be able to see around and control the joystick due to occlusion. Thus, a significant challenge here is to build caregiving robots that allow multiple modes of interfaces to control the robot. Participants also highlighted that dressing the lower body is more complex than the upper body. In most cases, lower body dressing is performed while the care recipient is lying on their bed. Dressing them in this state requires high payload capacity and reasoning under partial observability. These requirements make the physical interaction in this task particularly challenging. Also, bathing and dressing tasks would conceivably require a robot to collaborate with the user and their caregiver(s), and task allocation during physical caregiving is a major challenge.

Transferring. At first glance, though transferring may seem to require a robot with high payload capacity, our users highlight that existing robot hardware can still be used. This is possible by collaborating with human caregivers to operate assistive devices such as hoist lift, sit-to-stand lift, sliding board, etc. A major challenge is to design intelligent collaboration policies, and enable seamless integration between these robots and other assistive devices. As pointed out by a care recipient, transferring can begin with coarsely moving them to a bed or a wheelchair using an assistive device such as a hoist lift. The robot can then be used for finer limb-repositioning to successfully complete this ADL.

X. DISCUSSION

We introduced SPARCS, a framework for physical robot caregiving. SPARCS enables roboticists to translate real-world care requirements into guidelines for physical robot caregiving. Occupational therapists use similar frameworks when designing caregiving interventions. Compared to these frameworks, we define the components of SPARCS by grounding them in robotics. We release *SPARCS-box*, which allows roboticists to communicate with stakeholders from the caregiving community. In the future, we intend to improve *SPARCS-box* to make it more accessible and add features that incentivize stakeholders to use this platform.

SPARCS currently defines *Building Blocks* and *Structured Workflow* in a fairly subjective manner. Though we highlighted some possible ways of representing them, it remains to be explored what an ideal representation would be. Through this work, we are taking the first steps towards systematically structuring this impactful but scattered field of physically assistive robotics. We hope that our framework will speed up the progress in this domain, bringing us one step closer to caregiving robots that can provide long-term assistance.

XI. ACKNOWLEDGMENT

This work was funded by the NSF IIS (#2132846). We thank Daniel Stabile, Abrar Anwar, Boxin Xu and Advika Kumar for their help with experimental setup and figures.

REFERENCES

- [1] U. S. C. Bureau., “Americans with disabilities: 2014,” 2014.
- [2] Á. Graça, M. A. d. Nascimento, E. L. Lavado, and M. R. Garanhani, “Quality of life of primary caregivers of spinal cord injury survivors,” *Revista brasileira de enfermagem*, 2013.
- [3] L. E. Dreer, T. R. Elliott, R. Shewchuk, J. W. Berry, and P. Rivera, “Family caregivers of persons with spinal cord injury: Predicting caregivers at risk for probable depression,” *Rehab. Psychology*, 2007.
- [4] A. Chio, A. Gauthier, A. Vignola, A. Calvo, P. Ghiglione, E. Cavallo, A. Terreni, and R. Mutani, “Caregiver time use in als,” *Neurology*, 2006.
- [5] S. W. Brose, D. J. Weber, B. A. Salatin, G. G. Grindle, H. Wang, J. J. Vazquez, and R. A. Cooper, “The role of assistive robotics in the lives of persons with disability,” *AJPM&R*, 2010.
- [6] J. Broekens, M. Heerink, H. Rosendal *et al.*, “Assistive social robots in elderly care: a review,” *Gerontechnology*, 2009.
- [7] “SPARCS-box,” <https://emprise.cs.cornell.edu/sparcsbox>.
- [8] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, “Expressive body capture: 3d hands, face, and body from a single image,” in *CVPR*, 2019.
- [9] R. Ye, W. Xu, H. Fu, R. K. Jenamani, V. Nguyen, C. Lu, K. Dimitropoulou, and T. Bhattacharjee, “Rcareworld: A human-centric simulation world for caregiving robots,” in *IROS*, 2022.

- [10] P. Kadleček, A.-E. Ichim, T. Liu, J. Krivánek, and L. Kavan, “Reconstructing personalized anatomical models for physics-based body animation,” *TOG*, 2016.
- [11] H. Ryu, M. Kim, S. Lee, M. S. Park, K. Lee, and J. Lee, “Functionality-driven musculature retargeting,” in *Computer Graphics Forum*, 2021.
- [12] Y. Gao, H. J. Chang, and Y. Demiris, “User modelling for personalised dressing assistance by humanoid robots,” in *IROS*, 2015.
- [13] Y. Jiang and C. K. Liu, “Data-driven approach to simulating realistic human joint constraints,” in *ICRA*, 2018.
- [14] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, “Assistive gym: A physics simulation framework for assistive robotics,” in *ICRA*, 2020.
- [15] N. B. Reese and W. D. Bandy, *Joint range of motion and muscle length testing-E-book*. Elsevier Health Sciences, 2016.
- [16] N. Ciesla, V. Dinglas, E. Fan, M. Kho, J. Kuramoto, and D. Needham, “Manual muscle testing: a method of measuring extremity muscle strength applied to critically ill patients,” *JoVE*, 2011.
- [17] M. F. Folstein, S. E. Folstein, and P. R. McHugh, ““mini-mental state”: a practical method for grading the cognitive state of patients for the clinician,” *Journal of Psychiatric Research*, 1975.
- [18] J. Morley and N. Tumosa, “Saint louis university mental status examination (slums),” *Aging Successfully*, 2002.
- [19] W. H. Organization, *International Classification of Functioning, Disability, and Health: Children & Youth Version: ICF-CY*. World Health Organization, 2007.
- [20] G. Canal, C. Torras, and G. Alenyà, “Are preferences useful for better assistance? a physically assistive robotics user study,” *THRI*, 2021.
- [21] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, “Development and evaluation of a flexible interface for a wheelchair mounted robotic arm,” in *HRI*, 2008.
- [22] S. Javdani, H. Admoni, S. Pellegrinelli, S. Srinivasa, and J. Bagnell, “Shared autonomy via hindsight optimization for teleoperation and teaming,” *IJRR*, 2018.
- [23] S. Jain and B. Argall, “Probabilistic human intent recognition for shared autonomy in assistive robotics,” *THRI*, 2019.
- [24] H. J. Jeon, D. P. Losey, and D. Sadigh, “Shared autonomy with learned latent actions,” *arXiv preprint arXiv:2005.03210*, 2020.
- [25] G. Canal, G. Alenyà, and C. Torras, “A taxonomy of preferences for physically assistive robots,” *RO-MAN*, 2017.
- [26] —, “Adapting robot task planning to user preferences: an assistive shoe dressing example,” *Autonomous Robots*, 2019.
- [27] G. Yang, S. Wang, J. Yang, and P. Shi, “Desire-driven reasoning considering personalized care preferences,” *Transactions on Systems, Man, and Cybernetics: Systems*, 2021.
- [28] T. Bhattacharjee, M. E. Cabrera, A. Caspi, M. Cakmak, and S. S. Srinivasa, “A community-centered design framework for robot-assisted feeding systems,” in *ASSETS*, 2019.
- [29] “Unified robot description format (urdf),” <http://wiki.ros.org/urdf>, accessed: 2022-07-31.
- [30] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, “3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans,” *arXiv preprint arXiv:2002.06289*, 2020.
- [31] B. Scassellati, “Theory of mind for a humanoid robot,” *Autonomous Robots*, 2002.
- [32] J. Ondras, A. Anwar, T. Wu, F. Bu, M. Jung, J. J. Ortiz, and T. Bhattacharjee, “Human-robot commensality: Bite timing prediction for robot-assisted feeding in groups,” *arXiv preprint arXiv:2207.03348*, 2022.
- [33] D. Harel, “Statecharts: a visual formalism for complex systems,” *Science of Computer Programming*, 1987.
- [34] “SPARCS supplementary website,” <https://emprise.cs.cornell.edu/sparcs>.
- [35] D. Gallenberger, T. Bhattacharjee, Y. Kim, and S. S. Srinivasa, “Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding,” in *HRI*, 2019.
- [36] D. Berenson, S. Srinivasa, and J. Kuffner, “Task space regions: A framework for pose-constrained manipulation planning,” *IJRR*, 2011.
- [37] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, “Learning an animatable detailed 3d face model from in-the-wild images,” *ToG*, 2021.
- [38] R. Feng, Y. Kim, G. Lee, E. K. Gordon, M. Schmittle, S. Kumar, T. Bhattacharjee, and S. S. Srinivasa, “Robot-assisted feeding: Generalizing skewering strategies across food items on a plate,” in *ISRR*, 2019.