rrSDS 2.0: Incremental, Modular, Distributed, Multimodal Spoken Dialogue with Robotic Platforms

Anna Manaseryan[a], Porter Rigby[a], Brooke Matthews[a], Ryan Whetten[b], Catherine Henry[a], Josue Torres-Fonseca[c], Enoch Levandowsky[a], Casey Kennington[a]

[a]Department of Computer Science, Boise State University
[b]Laboratoire Informatique d'Avignon, Avignon Université
[c]Department of Computer Science, University of Michigan
Corresponding author: caseykennington@boisestate.edu

Abstract

This demo will showcase updates made to the 'robot-ready spoken dialogue system' built on the Retico framework. Updates include new modules, logging and real-time monitoring tools, integrations with the Coppelia Sim virtual robot platfrom, integrations with a benchmark, improved documentation, and pypi environment usage.

1 Introduction

The 'robot-ready spoken dialogue system' (rrSDS) was introduced in Kennington et al. (2020) based on the five requirements of **modularity**, **multimodality**, **distributiveness**, **incrementality** and **alignment**. Modularity ensures that the system components can be independently integrated. Multimodality allows the system to process various input forms. Distributiveness enables components to function across different systems. Incrementality processes inputs quickly in real-time. Alignment ensures that inputs from different modalities are synchronized.The rrSDS architecture continues to use Retico (Michael and Möller, 2019) which adopts the Incremental Unit Framework (Schlangen and Skantze, 2009) where incremental units (IUs) are passed between modules. rrSDS users can define their own IUs or use ones that are pre-defined. We report updates with new modules that maintain these requirements.

Upgraded Modules	New Modules
YOLOv11	RT-DETR
YOLOv12	SAM
MistyPy	DINOv2
MistyRobot	Whisper
Misty FaceDetection	Wav2vec
ObjectPermanence	HandTracking
	PoseTracking
	HuggingFace

Table 1: Summary of upgraded and newly added modules in rrSDS 2.0

2 Modules

In this section, we explain the perception, robot, simulation, language understanding and generation, and system-level modules that have been upgraded or newly added to form rrSDS 2.0. Table 1 summarizes the upgraded and newly added modules in rrSDS 2.0.

Object Detection & Feature Extraction YOLO models are known for their high detection accuracy with real-time inference speed. **YOLOv11** has a CNN-based architecture (Jocher and Qiu, 2024), while **YOLOv12** has an attention-centric design that achieves state-of-the-art object detection accuracy in real-time (Tian et al., 2025). rrSDS now includes both YOLOv11 and YOLOv12. We have also added the Real-Time DEtection TRansformer (**RT-DETR**) model, which uses a hybrid encoder and transformer decoder to perform object detection (Lv et al., 2024), and the Segment Anything Model (**SAM**) for object segmentation (object masking as well as bounding box information) without labels (Kirillov et al.), and a module for the **DINOv2** vision model, which differs from other models because its training is largely unsupervised, which may have better theoretical alignment for some robot tasks and systems (Oquab et al., 2023). All vision modules use the same IU types to make integration and replacement seamless.

Local Speech Recognition Two local automatic speech recognition (ASR) models have been added to rrSDS. **Whisper** is an ASR trained on 680k hours of multilingual and multitask data. The model uses an encoder-decoder transformer architecture (Radford et al., 2022). **Wav2vec** has been trained on 960 hours of unlabeled audio data. While other models, including Whisper, use spectrograms, Wav2vec uses raw audio as its input (Baevski et al., 2020). Prior work has evaluated Wav2vec's incremental ASR capabilities as part of

Proceedings of the 26th Annual Meeting of SIGDIAL, pages 637–640
Aug 25–27, 2025.

rrSDS (Whetten et al., 2023).

Pose Tracking Hand and body tracking modules use MediaPipe's Hand[1] and Pose[2] Landmarkers. These detect 21 hand and 33 body landmarks as 3D points (horizontal, vertical, depth).

The **HandTrackingModule** outputs 3D landmarks and left/right labels. The **PoseTrackingModule** outputs 3D body landmarks and a segmentation mask. Both modules allow configuration of model complexity, detection thresholds, video mode, smoothing (for pose), and the maximum number of detected hands (for hand tracking).

Misty II The Misty modules enable rrSDS to interface directly with the Misty robot through two components. **MistyPy** connects to the robot via its IP address using the Robot class and is initialized separately from standard rrSDS modules. **MistyRobot** provides access to robot functionalities, including receiving camera input and sending commands for movement and speech. The **FaceDetectionModule** detects human faces in single images or video streams and outputs bounding box coordinates. It requires a video input source, such as the VideoModule or the MistyRobot camera, and supports applications such as face recognition and gaze tracking.

Benchmarks The ALFRED benchmark for learning natural language mappings to action sequences (Shridhar et al., 2020) is actively being integrated into the rrSDS ecosystem, with the goal of providing evaluations based on real-time interactions. The ALFRED benchmark works from the perspective of the virtual robot and the transcription of what the human is instructing it to do. Additional benchmarks are also being investigated, with a focus on human-led interactions.

CoppeliaSim rrSDS now includes several modules aimed at interfacing with the CoppeliaSim robot simulator.[3] These modules include a general module for simulation of joint manipulation, a camera module from vision sensors internal to a CoppeliaSim simulation, a module for controlling a custom Cozmo robot within the simulation, and a module for receiving state updates from this custom Cozmo robot. The state and camera modules act as producing modules, while the general module and Cozmo controlling module both act as consuming modules. Together, these additions allow for real-time manipulation of robot simulations within the CoppeliaSim software via rrSDS, as shown in Figure 1.

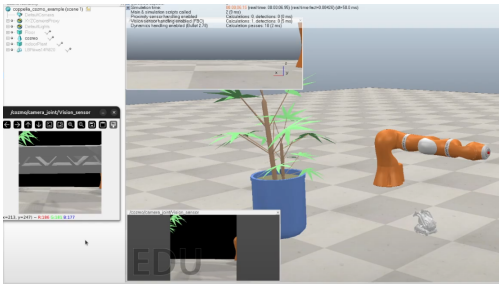

Figure 1: Cozmo robot integration in CoppeliaSim; a third-person and Cozmo view are visible as separate windows.

Cozmo SLAM & Object Permanence rrSDS supports an implementation of Simultaneous Localization and Mapping (**SLAM**) through a module for tracking object locations termed *object permanence* which leverages the 3D Viewer and Navigation Memory Map available through the Cozmo SDK. When an object has been perceived by Cozmo, that is, the robot has stopped moving and a frame from the camera has been passed to an object detection model, an object placeholder is inserted into Cozmo's Navigation Memory Map. The Navigation Memory Map is a quad-tree map that stores information about the robot's exploration space. The high-level purpose of this module is to provide Object Permanence to the robot when exploring and interacting with the physical world. Object permanence allows for systems where Cozmo can directly revisit perceived objects on request. Further details about this module as well as an example of how it might be leveraged in a rrSDS pipeline can be found in (Torres-Fonseca et al., 2022)

Language Understanding & Generation rrSDS supports real-time text generation using pre-trained language models. The **Retico-HuggingFace** module enables the generation of real-time text based on HuggingFace models. The module uses HuggingFace's pipeline API to integrate text generation models and supports interactive dialogue by generating responses based on recognized speech.

The original rrSDS had an incremental version of **RASA** (Bocklisch et al., 2017), but rrSDS 2.0

[1] https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker

[2] https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

[3] https://www.coppeliarobotics.com/

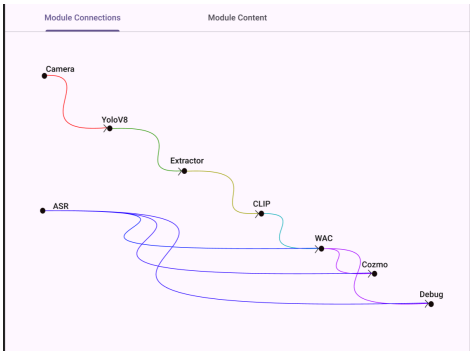

Figure 2: Visualization of module data flow.

has the more recent RASA 3.0 (which makes use of more recent language models for natural language understanding), which has been evaluated using incremental metrics in Whetten et al. (2023).

Visualization Tools The visualization tool provides real-time graphical feedback of module connections and data flow in an rrSDS runner. It streams information to a web interface, allowing the interface to run remotely. The tool shows a live module graph and the data passed between modules. Once the server, client, and server log module are running, the visualization updates automatically, helping with debugging and development. Figure 2 shows the visualization tool displaying the modules and how they are connected, and Figure 3 shows IU outputs from modules in realtime.

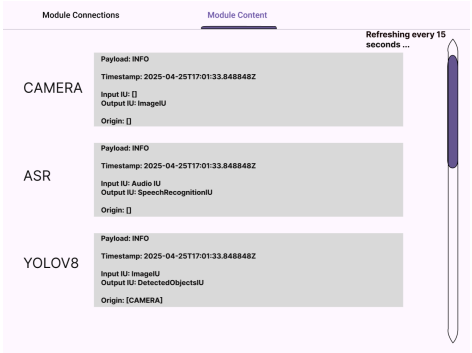

Figure 3: Visualization of IU flow between modules.

Logging The **SimpleLogger** module logs all IU types into a user-defined *.json* file. Logged information includes: when each IU was created, the payload or information they contain, their update type per the IU conceptual framework, and their module of origin. Any number of modules can be routed into the logger, and the resulting log file exported to common data analysis tools.

3 Example System Implementations

In this section, we present some systems that demonstrate different usages of modalities in rrSDS.

Cozmo on CoppeliaSim We offer an example system for controlling a virtual Cozmo robot in CoppeliaSim, depicted in Figure 1. The full system uses an ASR module, and three CoppeliaSim modules that process the virtual environment, Cozmo's internal states, Cozmo's virtual camera, Cozmo's actions, and debugging. The system is simple in that it is built as a "verbal joystick" (e.g., *move forward, stop, turn right, turn left*).

Nim Game In this example, Misty II robot plays the game of Nim with a user using reinforcement learning.[4] The rrSDS Yolov8 module uses image information from Misty's camera module to find objects in the image and a game master modules count objects and track the game state in real-time and gives feedback by speaking.

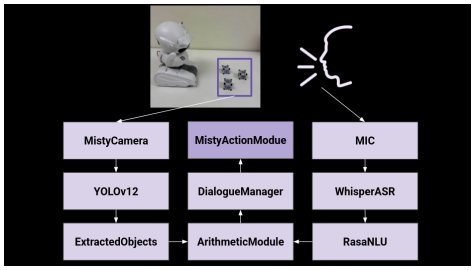

Figure 4: rrSDS Interaction Flow for Arithmetic Learning with Misty II.

Math Tutor In this system, the Misty II robot acts as an interactive tutor to teach young learners basic concepts such as numbers, basic arithmetic operations, and colors. The system utilizes Misty's camera along with several rrSDS modules to create real-time, multimodal interactions. Different modules guide the robot in helping children count objects, perform simple addition and subtraction, and recognize colors through object detection. Figure 4 illustrates the system pipeline in rrSDS, where the Misty robot engages the user in an arithmetic task. Misty prompts the user to add or remove a certain number of cubes placed in front of it, then asks how many cubes remain. It processes the user's spoken response and simultaneously uses object detection to verify whether the number of visible cubes matches the expected answer.

[4] https://github.com/bjBSU/nim

4 Conclusion & Future Work

rrSDS 2.0 builds on the original rrSDS, increasing the number of supported modules, ability to expand new modules for users, example systems, and research support tools. It continues to be integrated with psi (Bohus et al., 2017) and the Robotics Operating System.

While integrating rrSDS 2.0 across physical and simulated robots, we faced challenges balancing real-time, multimodal interaction.

In the future, we are exploring integrations with other benchmarks and the Remdis incremental processing framework for language models (Chiba et al., 2024).

Acknoledgements We would like to thank the anonymous reviewers for their helpful feedback. This material is based upon work supported by the National Science Foundation under Grant No. 2140642 & 2343118. Finally, we would like to thank Beth Grenz for her help with human participants.

References

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv [cs.CL]*.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *Proceedings of the 31st Conference on Neural Information Processing Systems*.

Dan Bohus, Sean Andrist, and Mihai Jalobeanu. 2017. Rapid development of multimodal interactive systems: a demonstration of platform for situated intelligence. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, New York, NY, USA. ACM.

Yuya Chiba, Koh Mitsuda, Akinobu Lee, and Ryuichiro Higashinaka. 2024. The remdis toolkit: Building advanced real-time multimodal dialogue systems with incremental processing and large language models. In *Proc. IWSDS*, pages 1–6.

Glenn Jocher and Jing Qiu. 2024. Ultralytics YOLO11.

Casey Kennington, Daniele Moro, Lucas Marchand, Jake Carns, and David McNeill. 2020. rrSDS: Towards a robot-ready spoken dialogue system. In *Proceedings of the 21st Annual SIGdial Meeting on Discourse and Dialogue*, Virtual. Association for Computational Linguistics.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304. 02643*.

Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. 2024. RT-DETRv2: Improved baseline with bag-of-freebies for real-time DEtection TRansformer. *arXiv [cs.CV]*.

Thilo Michael and Sebastian Möller. 2019. ReTiCo: An open-source framework for modeling real-time conversations in spoken dialogue systems. In *Tagungsband der 30. Konferenz Elektronische Sprachsignalverarbeitung 2019*, ESSV, pages 134–140, Dresden. TUDpress, Dresden.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2023. DINOv2: Learning robust visual features without supervision.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

David Schlangen and Gabriel Skantze. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 710–718, Athens, Greece. Association for Computational Linguistics.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks.

Yunjie Tian, Qixiang Ye, and David Doermann. 2025. YOLOv12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502. 12524*.

Josue Torres-Fonseca, Catherine Henry, and Casey Kennington. 2022. Symbol and communicative grounding through object permanence with a mobile robot. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 124–134, Edinburgh, UK. Association for Computational Linguistics.

Ryan Whetten, Enoch Levandovsky, Mir Tahsin Imtiaz, and Casey Kennington. 2023. Evaluating automatic speech recognition and natural language understanding in an incremental setting. In *Proceedings of the 27th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Maribor, Slovenia. SEMDIAL.