June 2026

Generating Synthetic Multi-Turn Conversations For Scalable Functional Testing Of Conversational Ai Systems

Niklas Weller
University of St. Gallen, niklas.weller@unisg.ch

Shijing Cai
Calvin Risk AG, sc@calvin-risk.com

Syang Zhou
Calvin Risk AG, sz@calvin-risk.com

GENERATING SYNTHETIC MULTI-TURN CONVERSATIONS FOR SCALABLE FUNCTIONAL TESTING OF CONVERSATIONAL AI SYSTEMS

Short Paper

Niklas Weller, University of St.Gallen, St.Gallen, Switzerland, niklas.weller@unisg.ch

Shijing Cai, Calvin Risk AG, Zurich, Switzerland, sc@calvin-risk.com

Syang Zhou, Calvin Risk AG, Zurich, Switzerland, sz@calvin-risk.com

Abstract

Artificial intelligence (AI) systems with conversational interfaces are increasingly used to augment humans and even automate complex workflows. Yet systematic functional testing remains difficult because of open-ended multi-turn input. Existing evaluation benchmarks focus on isolated single-turn performance and provide limited insight into how systems behave in realistic interaction scenarios. This paper investigates how synthetic multi-turn conversations can be generated to support scalable functional testing of conversational AI systems. Using a design science research approach, we develop an artifact that employs LLM-based role play to generate synthetic conversations for testing an insurance claim decision system. The generation process combines software testing techniques such as equivalence partitioning and boundary value analysis with persona-based user simulation. We report preliminary results from the first design cycle and derive six design principles for generating diverse, faithful, and diagnostically useful conversational test data.

Keywords: Synthetic Data, Testing, LLM Role Play, Conversational AI, Automation

1 Introduction

Since the launch of ChatGPT in November 2022, conversational artificial intelligence (AI) systems have exploded in popularity. Enabled by large language model (LLM) technology, users could easily convey complex information in natural language. Today, these maturing systems promise valuable workflow automations through agentic capabilities (Purdy, 2024). Nevertheless, they still often rely on conversational input. The open-endedness and the multi-turn characteristics of this input type pose a challenge for testing (Ganguli et al., 2022; Heimburg et al., 2025; Ma et al., 2024; Magalhaes et al., 2025). Evaluating input-output relationships for every possible input is infeasible. In practice, this often causes these systems to be solely evaluated through unsystematic and expensive manual testing (Magalhaes et al., 2025) or through benchmarks isolated from the application context (Ni et al., 2025). Thus, there is a need for a scalable approach to strategically generate multi-turn conversations as test cases to comprehensively assess the real-world performance of conversational AI systems.

Synthetic data has shown promises for scalable testing in multiple fields. For example, in financial fraud detection, where actual fraudulent transaction data is scarce, it allows for assessing the capabilities of detection technologies by emulating fraudulent cases (Fiore et al., 2019). We suggest a similar approach for the problem of open-ended multi-turn conversations as system input. Consequently, we formulate the following research question: *What design principles should guide the generation of synthetic multi-turn conversations for functional testing of conversational AI systems?*

To address the research question, we rely on design science research methodology. In an iterative process respecting industry requirements, we develop an artifact to generate multi-turn conversations as test cases. Our approach is grounded in established software testing techniques and utilizes state-of-the-art LLM technologies for role-playing human users. We make the following contributions to design knowledge:

1. We contribute an instantiated artifact for generating synthetic conversations for scalable functional testing of a coverage decision automation for insurance companies. An example *system under test* is illustrated in Figure 1. While also relevant for testing open-ended LLM outputs, we focus on testing discrete insurance coverage decisions with our test case generator artifact. The resulting conversations can be applied for testing decisions on unit level (e.g. LLM2 in Figure 1 as unit under test) or on system level (e.g. full multistep decision system under test as illustrated in Figure 1).
2. We contribute six design principles as high-level guidance of form and function for generating conversational test data (see Section 4.1). They are grounded in industry requirements as well as in literature on software testing and synthetic data generation. Applying these principles should yield test data that is diverse, faithful, and diagnostically useful for testing conversational AI systems at scale.

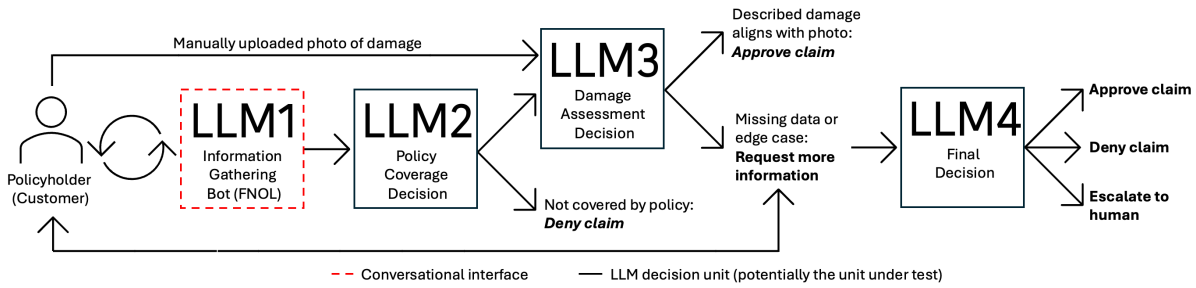

Figure 1. Example system under test instantiating a simplified workflow automation of an insurance claim decision. The multi-turn conversation between the policyholder and LLM1 is generated by our artifact as it provides the relevant context for the decision.

2 Related Work

2.1 Conversational AI Evaluation & Software Testing

The LLMs enabling modern conversational AI systems are commonly evaluated using benchmark datasets that assess their performance in isolation (Ni et al., 2025). While useful to compare general capabilities and track progress over time, these evaluations remain largely static and limited to single-turn exchanges (Ma et al., 2024), despite the need for scenario-oriented evaluation methods (Ni et al., 2025). As a result, they provide incomplete and inadaptable coverage of the diverse and evolving contexts in which conversational systems operate.

This isolated view is particularly problematic given that LLMs are increasingly deployed as core part of software systems – undeterred by their inscrutability producing unpredictable behaviors in deployment (Ganguli et al., 2022; Heimburg et al., 2025). Thus, we need to approach dynamic testing that respects LLMs as component of a specific software system and its sociotechnical environment (Kastner, 2025). Currently, this is often done through exploratory and ad hoc manual annotation (Magalhaes et al., 2025).

Functional software testing provides strategies to approach testing of black-box systems by assessing the system's intended behavior rather than its internal structure. For dealing with the open-ended input, the testing techniques of input space partitioning, more specifically equivalence partitioning and boundary value analysis offer useful guidance (Ammann & Offutt, 2016; Beizer, 1995). Black-box testing has already been successfully deployed for natural language processing models, previous to the LLM era (Ribeiro et al., 2020). By combining the established techniques with LLM technology for

generating synthetic conversations, we plan to advance testing of conversational AI systems in their specific context.

2.2 Synthetic Data Generation with LLMs

To achieve targeted testing of conversational AI systems, we require dynamic test cases that respect the context of the software system and its broader environment. As these datasets usually not exist pre-deployment, it can only be approached through substitution with synthetic conversational data. In other fields, the usage of synthetic data for testing has been successfully applied (e.g. Fiore et al., 2019). Research has explored the possibility of utilizing LLMs for synthetic conversation generation to obtain scalable context-specific dialogue data. By role-playing humans (Shanahan et al., 2023), relatively cheap and scalable user inputs can be simulated (Chen et al., 2024; Long et al., 2024; Park et al., 2024). Frameworks like UGRO, USimAgent, and SimUSER effectively employ LLMs to simulate lifelike interactions (Hu et al., 2023; Xiang et al., 2024; Zhang et al., 2024). While such methods show promises in increasing data volume and scenario variety, they are not generated for the purpose of testing conversational systems. To achieve this, we need to combine findings from studies using LLMs for generating synthetic conversations with the properties of high-quality test cases according to software testing principles.

3 Methodology

We followed a design science research (DSR) methodology based primarily on Peffers et al. (2007) to iteratively develop and refine the synthetic conversation generator. Specifically, this short paper reports the first design cycle spanning problem identification, definition of objectives, design and development, and initial demonstration. We use Hevner et al. (2004) as a complementary lens to motivate the relevance of the problem environment and the utility-oriented nature of the artifact. In addition, Gregor and Hevner (2013) guide the positioning of our contribution as an instantiated artifact and nascent design knowledge in the form of design principles. In line with DSR's purpose of creating a useful artifact while contributing generalizable design knowledge, the author team included both academic researchers and practitioners familiar with the problem. The latter hold executive and industry research positions in a company providing automated testing solutions for AI systems. Over a six-month period, weekly meetings and informal design discussions served as opportunities to explore current challenges, assess potential solution directions, and iteratively refine problem understanding – simultaneously we developed the artifact. The emerging design requirements as listed in Figure 2 synthesize the practical insights with theoretical background from software testing (Ammann & Offutt, 2016) and synthetic data objectives (Long et al., 2024). The close involvement of industry representatives ensured the relevance and practical grounding of the requirements, while theoretical framing guided abstraction and the development of more generalizable design principles.

We report the preliminary results of the first design cycle in this short paper (problem, objectives, development, demonstration). The evaluation is multi-step and – thus far – only partly completed. Based on the full evaluation results we will iteratively refine the synthetic conversation generator and apply it to different contexts in the upcoming design cycles.

4 Preliminary Results

A problem can be regarded as the difference between the *goal state* and the *current state* (Hevner et al., 2004). Ideally, we would test every possible input and observe the outcome (*goal state*). Currently, conversational AI is mostly tested with unstructured and expensive human annotation (*current state*, Magalhaes et al., 2025). We formulate the requirements based on the objective to move closer to the goal state by strategically approximating the input space with synthetic data generation grounded in software testing techniques. Figure 2 provides an overview over the derived design requirements (DRs), design principles (DPs), and design features (DFs). Because these were derived in a policy-grounded insurance setting with discrete decision outcomes, their transfer to other conversational AI systems may require adaptation.

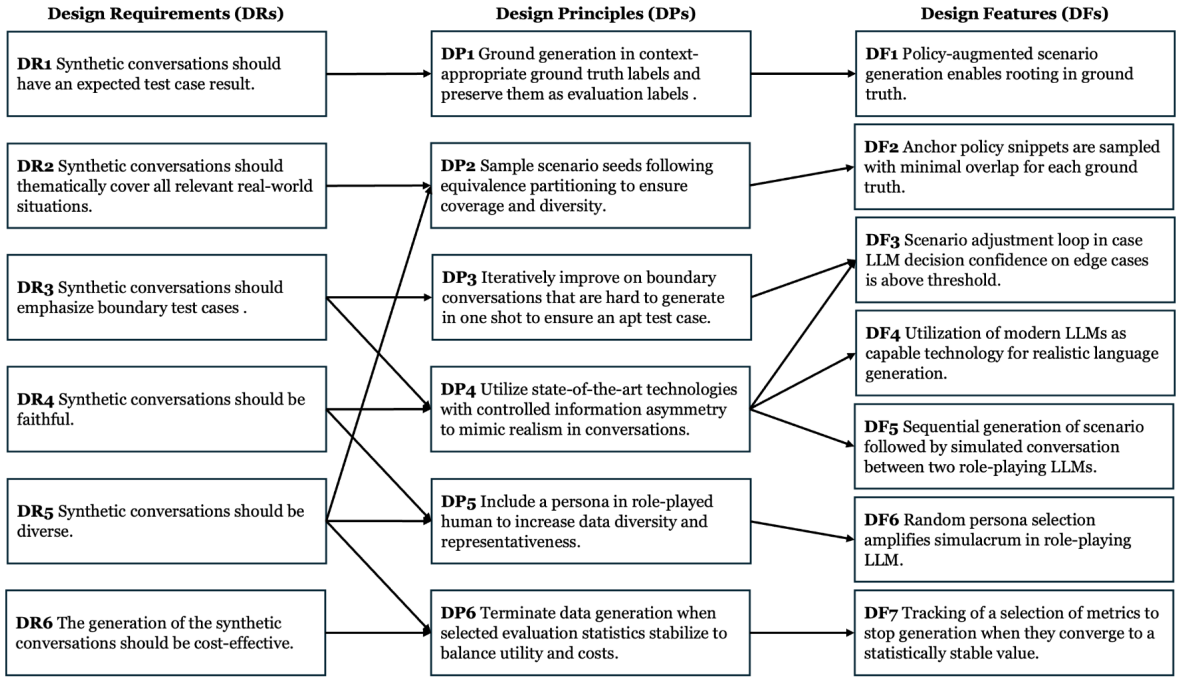

Figure 2. Overview of the design requirements, principles, features, and relations.

4.1 Design Principles

In line with Jones and Gregor (2007), our design principles are high-level design rules of form and function. They are formulated in a way to provide concrete guidance to designers of multi-turn conversation generators for functional testing and cover the preliminary results of our iterative development. While our specific setting focuses on testing discrete LLM decisions based on a policy-document, the design principles should abstract to generating multi-turn conversations as test cases for any conversational AI system.

DP1: Ground generation in context-appropriate ground truth labels and preserve them as evaluation labels. The first design principle is based on the design requirement that synthetic conversations for testing should have an expected result. This, in turn, is based on the necessary elements composing a test case for complete execution and evaluation (Ammann & Offutt, 2016), which include the test case values (conversations) and the expected result (ground truth to evaluate against). The expected result or ground truth depends on the specific testing target and respecting the level of testing (e.g. unit testing of LLM2 decision or system testing of multiple interrelated components).

An important boundary condition posits that an expected result can be determined. While this is trivial on tasks with a verifiably correct result, the expected result will often depend on the sociotechnical environment in which the conversational AI system is deployed. In practice, this means that the expected result of the test case should specify the normatively correct solution for the given context. In our insurance example of Figure 1, the leniency of approving an edge case claim can vary between different organizations due to different policy interpretations or company goals.

DP2: Sample scenario seeds following equivalence partitioning to ensure coverage and diversity. For deployment, practitioners demand estimates over how the system behaves in all real-world situations. This intuitive requirement can be implemented by following the testing technique of input space partitioning, more specifically equivalence partitioning (Ammann & Offutt, 2016; Myers et al., 2004). While the open-ended inputs prevent testing everything, we can partition the input parameters into regions and then choose at least one value from each region. In our case, these parameters are the ground truth and the anchor policy snippets. This forces the generated conversations to cover relevant

claim outcomes and topics. The equivalent coverage of the input partitions also helps increasing content variability and thereby diversity, which – together with faithfulness – is a major challenge in synthetic data generation (Long et al., 2024).

DP3: Iteratively improve on conversations that are hard to generate in one shot to ensure an apt test case. This principle is inspired by the testing technique of boundary value analysis (Ammann & Offutt, 2016). As errors often occur on the edge of the partitions (see DP2), test cases on the boundaries can provide valuable diagnostic insights. Thus, we put an emphasis on generating high quality boundary cases, which LLMs can struggle to generate due to the involved levels of subjectivity (Li et al., 2023).

A hard-to-generate case in our setting is a claim that sits near the boundary between covered and not covered because the same loss can be framed in multiple ways. For instance, a customer may report water damage to a device, while the policy covers accidental liquid contact but may exclude cases in which the facts suggest the damage was not caused by an unexpected and unintentional event. A useful boundary conversation must reveal enough detail to make both interpretations initially plausible. Because one-shot generation often fail to produce these nuances, we iteratively regenerate such scenarios until an auxiliary model's decision confidence falls below a predefined threshold.

DP4: Utilize state-of-the-art technologies with controlled information asymmetry to mimic realism in conversations. To generate faithful dialogue, we should rely on the best technology available to generate natural language and use multi-step approaches facilitating realistic information asymmetry (Long et al., 2024). As of writing, LLMs are a promising and prominently applied technology for synthetic data generation (Kim et al., 2025; Long et al., 2024). The capability to produce faithful text has been demonstrated multiple times with recent studies claiming that models even pass the Turing Test (Jones & Bergen, 2025; Turing, 1950) – a long standing challenge of language generation and imitation. On top of using a capable technology, we also shield information from the role-playing LLM to enhance realism. This is achieved by generating the scenario in a first step and afterwards providing only selected scenario components (the intention and some case-specific details) to the role-playing LLM, while withholding other metadata (ground truth, scenario reasoning, relevant policy snippets). This way, the actual information asymmetry of a real-world conversation is retained.

DP5: Include a persona in role-played human to increase data diversity and representativeness. LLM characteristics can be regarded as a superposition of simulacra (Shanahan et al., 2023). By prompting the model with a representative persona, a representative simulacrum for role-playing can be amplified. This allows the LLM to emulate distinct user types and communicative styles – such as cooperative, neutral, or resistant – addressing the diversity requirement (Long et al., 2024) of the synthetic conversations. Specifically, this technique allows for generating linguistically and behaviorally diverse conversations, even if they are part of the same equivalence partitioning region (DP2) and thus similar in content. The principle is testable by comparing persona-conditioned and non-persona-conditioned generation with respect to linguistic diversity, behavioral diversity, and downstream fault detection.

DP6: Terminate data generation when selected evaluation statistics stabilize to balance utility and costs. This principle addresses the industry requirement that the generation of test cases should be cost-effective. To achieve this, it draws on generalizability theory for informing when enough data has been generated. It posits that reliable measurements should remain stable across equivalence partitions (Cronbach et al., 1972). This is the only principle of Figure 2, which is yet to be implemented in our artifact. We plan on tracking metric stability throughout batches of generated conversations and stopping when results converge (DF7), indicating sufficient generalizability while avoiding unnecessary computation. To identify relevant metrics to track, we will conduct an initial evaluation.

4.2 Artifact

Figure 3 provides an overview over the synthetic conversation generation process, as implemented in our artifact. In a first step the scenario is generated. To achieve this, the generation instructions for a LLM are augmented with the full policy and a ground truth giving context to the scenario (DF1). Additionally, the instructions specify to ground the scenario situation on a sampled anchor snippet to

ensure coverage in line with equivalence partitioning (DF2). Once the scenario has been generated, the necessary information is, in a second step, provided to another LLM which is role-playing the human customer (DF4, DF5). On top of the necessary scenario information, the role-playing LLM is given a persona to adhere to (DF6) in order to amplify a representative simulacrum (Shanahan et al., 2023). We randomly sample a persona (aged 18+) from a collection of 1,586 personas representing the US census data and including personality traits (Castricato et al., 2025). The turn-taking dialogue then takes place between the role-played customer and the conversational AI system – in our case the interface is the first notice of loss (FNOL) insurance bot of the system represented in Figure 1. With a special token, the role-played customer can abandon the conversation. The final generated multi-turn conversation is then saved and enriched with the scenario metadata (e.g. ground truth, scenario reasoning, relevant policy snippets), completing the properties of a test case. If the generated conversation is intended to be an edge case in line with boundary value analysis, it is passed to another LLM forced to make a covered/not covered decision. In case the decision confidence – measured at the output probability distribution over the vocabulary – is above a threshold (e.g. above 80% probability of covered), the scenario is adjusted and the conversation is recreated until the threshold is met (DF3).

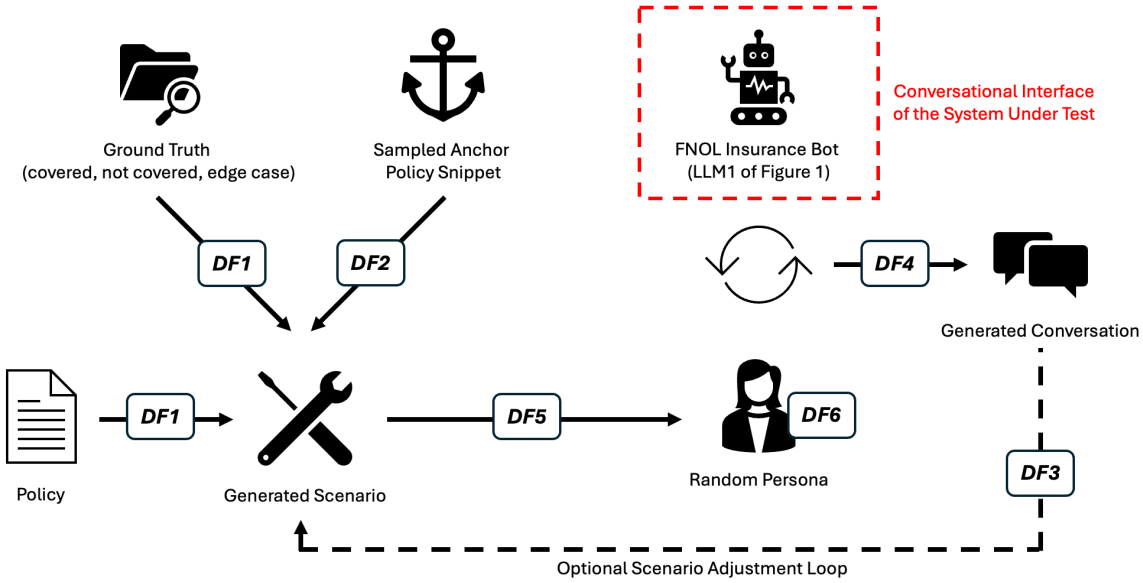

Figure 3. Illustration of the implemented design features and the resulting generation pipeline.

Figure 4 shows screenshots of the user interface implemented for visual inspection of the generated conversations. In line with the demonstration stage of the design cycle (Peffers et al., 2007), we can use it to show that the system works. For the evaluation step, we need to assess the quality of the generated conversations for testing purposes. The right panel of Figure 4 shows an initial automatic evaluation by visualizing the semantic coverage and providing some summary metrics for the scenario and the conversations.

Further details on the implementation and sample generations are provided in the artifact repository which we make available on GitHub[1].

[1] Link: *github.com/nwllr/Synthetic_Conversations_ECIS2026*

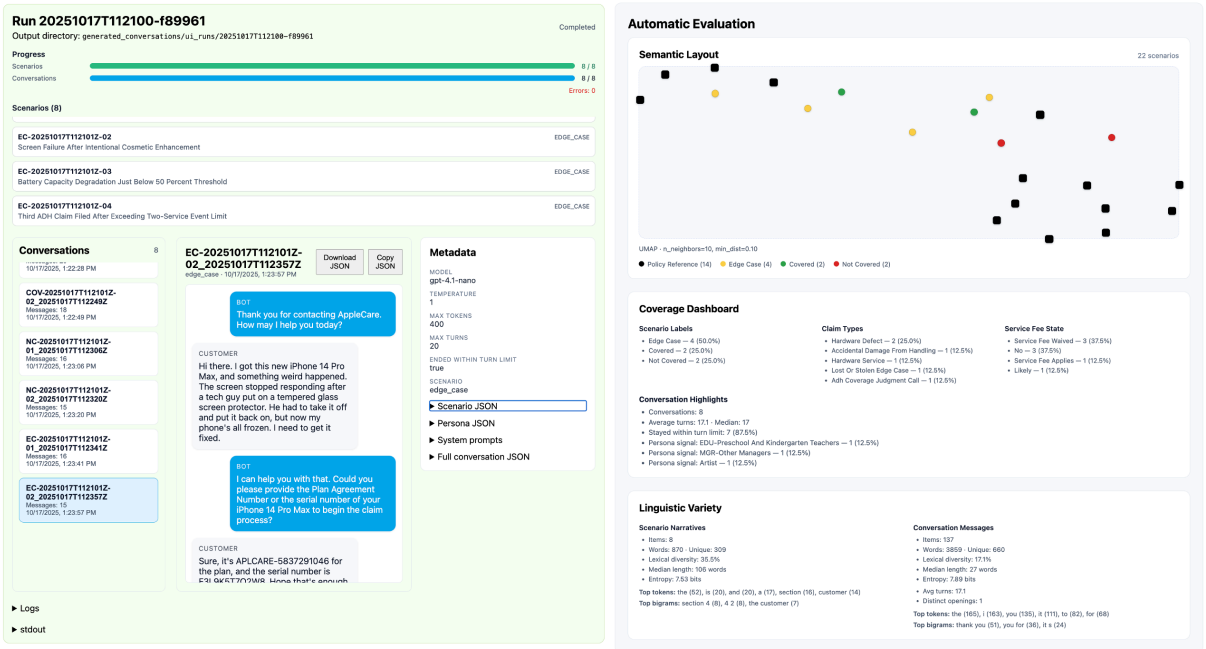

Figure 4. Screenshots from the user interface showing the conversation inspector (left) and an initial panel for automated evaluation measures (right).

5 Outlook

For completing the first design cycle, we plan on applying a multi-step evaluation to assess the quality of the generated conversation as test cases. Further, we aim to get insights into the effectiveness of each design principle to improving test case quality. We intend to rely on a combination of intrinsic/direct and extrinsic/indirect evaluation strategies (Long et al., 2024). For intrinsic evaluation of the data diversity, we will mostly utilize automated metrics assessing linguistic and semantic variety. Thus far, we have implemented some basic metrics for linguistic variety and a visualization for assessing semantic coverage (see Figure 4). For intrinsic evaluation of data faithfulness, we plan to rerun an experiment set up in a different context to see whether humans can differentiate the synthetic dialogues from actual dialogues. Additionally, we can judge faithfulness by assessing the appropriateness of the generated conversations for testing purposes via metrics on partition coverage (Ammann & Offutt, 2016). For extrinsic evaluation, we plan to set up an experimental evaluation to see whether the generated conversations are faithful to their testing purpose. Specifically, we will conduct an online experiment where human participants are tasked with providing a confidence score for a covered/not covered decision based on the policy. This way, the faithfulness of the conversations to the ground truth can be validated. Additionally, we will get an estimate for the effectiveness of the generated boundary conversations (here we hypothesize low and varied confidence in human participants). These evaluation steps will conclude our first design cycle.

We contribute an instantiated research artifact, which is publicly accessible in a GitHub repository. The setup includes a web user interface for easy manual inspection and initialization of generation runs. The pipeline can be adopted to other insurance policies. It can also serve as a basis for multi-turn data generation in other domains where a decision grounding document is available, e.g. for legal cases. Additionally, we contribute six design principles. These can be regarded as a nascent design theory representing a *level 2* contribution to design knowledge (Gregor & Hevner, 2013).

Naturally, our approach has limitations. Importantly, we presuppose full access to the conversational interface of the *system under test* as the required counterpart for the role-playing LLM. Moreover, for our artifact, we focus on a very specific setting of insurance coverage decisions. While the principles are formulated in a way that abstract from this specific setting, they might not apply to other settings

without modification. Additionally, other inputs – beyond the scope of this paper – might influence the outcome and could therefore be added to the test case generation (these include instructions, decision option descriptions, or multimodal input like the photo for LLM3 in Figure 1).

In the upcoming design cycle, we aim to address these limitations by assessing multi-turn conversation generation for testing a conversational AI system in a different setting. By validating, falsifying or extending the principles in another context, we will learn about the extent of their generalizability. In parallel, we are conducting experiments to gain a more nuanced view of *how* the different features of our generator system are affecting the output conversation quality.

Ultimately, we encourage cumulative efforts to extend design knowledge by building on the design principles derived in this study for creating test cases for conversational AI systems in any settings.

6 Acknowledgements

This work was supported by the Swiss Innovation Agency Innosuisse as part of project 129.524 INNO-ICT.

References

Ammann, P., & Offutt, J. (2016). *Introduction to Software Testing* (2 ed.). Cambridge University Press.

Beizer, B. (1995). *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. John Wiley & Sons, Inc.

Castricato, L., Lile, N., Rafailov, R., Fränken, J.-P., & Finn, C. (2025, January). PERSONA: A Reproducible Testbed for Pluralistic Alignment. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, & S. Schockaert, *Proceedings of the 31st International Conference on Computational Linguistics* Abu Dhabi, UAE.

Chen, J., Wang, X., Xu, R., Yuan, S., Zhang, Y., Shi, W., Xie, J., Li, S., Yang, R., Zhu, T., Chen, A., Li, N., Chen, L., Hu, C., Wu, S., Ren, S., Fu, Z., & Xiao, Y. (2024). From Persona to Personalization: A Survey on Role-Playing Language Agents. *Trans. Mach. Learn. Res.*, *2024*.

Cronbach, L. J., Gleser, G. C., Nanda, H., & Rajaratnam, N. (1972). *The Dependability of Behavioral Measurements. Theory of Generalizability for Scores and Profiles.* Wiley, New York.

Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection. *Information Sciences*, *479*, 448-455.

Ganguli, D., Hernandez, D., Lovitt, L., Askell, A., Bai, Y., Chen, A., Conerly, T., Dassarma, N., Drain, D., Elhage, N., Showk, S. E., Fort, S., Hatfield-Dodds, Z., Henighan, T., Johnston, S., Jones, A., Joseph, N., Kernian, J., Kravec, S.,…Clark, J. (2022). *Predictability and Surprise in Large Generative Models*. Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea.

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337–356.

Heimburg, V., Schreieck, M., & Wiesche, M. (2025). Complementor Value Co-Creation in Generative AI Platform Ecosystems. *Journal of Management Information Systems*, *42*(2), 491-528.

Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, & Sudha. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75-105.

Hu, Z., Feng, Y., Luu, A. T., Hooi, B., & Lipani, A. (2023). *Unlocking the Potential of User Feedback: Leveraging Large Language Model as User Simulators to Enhance Dialogue System*. 32nd ACM Int. Conference on Information and Knowledge Management, Birmingham, United Kingdom.

Jones, C. R., & Bergen, B. K. (2025). Large Language Models Pass the Turing Test. *arXiv preprint arXiv:2503.23674*.

Jones, D., & Gregor, S. D. (2007). The Anatomy of a Design Theory. *J. Assoc. Inf. Syst.*, *8*, 19.

Kastner, C. (2025). *Machine Learning in Production: From Models to Products*. MIT Press.

Kim, S., Suk, J., Yue, X., Viswanathan, V., Lee, S., Wang, Y., Gashteovski, K., Lawrence, C., Welleck, S., & Neubig, G. (2025, July). Evaluating Language Models as Synthetic Data Generators. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* Vienna, Austria.

Li, Z., Zhu, H., Lu, Z., & Yin, M. (2023). Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations. *arXiv preprint arXiv:2310.07849*.

Long, L., Wang, R., Xiao, R., Zhao, J., Ding, X., Chen, G., & Wang, H. (2024). On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey. In L.-W. Ku, A. Martins, & V. Srikumar, *Findings of the Association for Computational Linguistics (ACL)* Bangkok, Thailand.

Ma, C., Zhang, J., Zhu, Z., Yang, C., Yang, Y., Jin, Y., Lan, Z., Kong, L., & He, J. (2024). Agentboard: An Analytical Evaluation Board of Multi-turn LLM Agents. *arXiv preprint arXiv:2401.13178*.

Magalhaes, C., Santos, I., Stuart-Verner, B., & Santos, R. D. S. (2025). Testing the Untestable? An Empirical Study on the Testing Process of LLM-Powered Software Systems. 2025 IEEE International Conference on Source Code Analysis & Manipulation (SCAM),

Myers, G. J., Badgett, T., Thomas, T. M., & Sandler, C. (2004). *The Art of Software Testing* (Vol. 2). Wiley Online Library.

Ni, S., Chen, G., Li, S., Chen, X., Li, S., Wang, B., Wang, Q., Wang, X., Zhang, Y., & Fan, L. (2025). A survey on large language model benchmarks. *arXiv preprint arXiv:2508.15361*.

Park, J. S., Zou, C. Q., Shaw, A., Hill, B. M., Cai, C., Morris, M. R., Willer, R., Liang, P., & Bernstein, M. S. (2024). Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109*.

Peffers, K., Tuure, T., A., R. M., & and Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45-77.

Purdy, M. (2024). What Is Agentic AI, and How Will It Change Work? *Harvard Business Review*.

Ribeiro, M. T., Wu, T., Guestrin, C., & Singh, S. (2020). Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* Online.

Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role Play With Large Language Models. *Nature*, *623*(7987), 493-498.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, *59*(236), 433-460.

Xiang, W., Zhu, H., Lou, S., Chen, X., Pan, Z., Jin, Y., Chen, S., & Sun, L. (2024). *SimUser: Generating Usability Feedback by Simulating Various Users Interacting with Mobile Applications*. Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA.

Zhang, E., Wang, X., Gong, P., Lin, Y., & Mao, J. (2024). *USimAgent: Large Language Models for Simulating Search Users*. Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington DC, USA.