Chapter 3: Research Method

Nate Bachmeier

DIS-9991: Components of a Dissertation

January 16, 2022

Northcentral University

# Research Method

Design-science is a standard methodology for researching Information Technology (IT) problems. Hevner et al. (2004) propose a collection of guidelines for implementing this methodology (see Table 1). There are three phases to implementing this process. First, the researcher(s) must identify a domain-specific challenge. Next, that researcher creates artifacts that study this phenomenon. Third, those artifacts assess the topic and communicate answers to the research questions.

Table 1: Design-science Guidenlines (Hevner et al. 2004)

|  |  |
| --- | --- |
| Guideline | Description |
| Design as an Artifact | Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation. |
| Problem Relevance | Design-science research aims to develop technology-based solutions to important and relevant business problems. |
| Design Evaluation | A design artifact's utility, quality, and efficacy must rigorously demonstrate well-executed evaluation methods. |
| Research Contributions | Effective design-science research must provide transparent and verifiable contributions to design artifacts, foundations, and/or design methodologies. |
| Research Rigor | Design-science research relies on rigorous methods to construct and evaluate the design artifact. |
| Design as a Search Process | The search for a compelling artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment. |
| Communication of Research | Design-science research must be presented effectively both to technology-oriented and management-oriented audiences. |

This dissertation employs this methodology to improve special needs and elderly care with AI/ML and CV applications. Scalability, security, and privacy challenges prohibit studying this topic through traditional means. People are generally unwilling to undergo 24/7 video monitoring and disclose their most intimate conversations in the name of science. Future research needs to address those concerns. Meanwhile, this effort provisions industry-standard physics simulation environments to examine those interactions. Next, this project creates virtual devices (e.g., IP cameras) to extract a subject’s behavior and respond accordingly. Third, a data telemetry collection pipeline will assess the performance of virtual devices within a simulated world.

## Research Methodology and Design

### Artifact Driven Approach

Artifacts are a principal component of the constructive design methodology. This constructive research project will simulate human activity and then predict the subject’s intents using video streams. The solution uses open source tooling, enabling other researchers to extend these efforts for their projects. Derived works could include social studies, game theory, film production, and other fields of study. These diverse researchers need activity recognition capabilities across industry-standard video formats (e.g., RGB+D). Also, future researchers will need to extend the action-space taxonomy to support domain-specific intents. For instance, this research project does not support observing a soccer match. Though, this project should expose primitives for adding those future requirements.

### Runtime System Design

This research project includes subsystems for simulating human movements, observing those behaviors, extracting intents, reacting through CPS systems, and evaluating prediction accuracies (Figure 1). An experiment begins with a test-case specification that describes the scene, actors, animations, and virtual devices. First, the Runtime Environment Pipeline simulates the scene requirements while virtual IP-Cameras monitors and reacts appropriately. Next, the Feedback Monitoring Pipeline Telemetry persists prediction history into a time-series database. Lastly, an evaluation process can compare the test-case definition against the Decision History Store to assess the system’s performance.

Figure 1: Experiment DesignDiagram, schematic

Description automatically generated

#### Test Case Definition

A test case encapsulates a specific experiment. An arbitrary number of subjects will perform pre-configured animation sequences during the experiment, such as walking or failing. These behaviors occur within a dynamic world that supports typical real-world transforms. For example, the subject can turn off a light, move furniture, not modify the floor plan.

#### Data Generation Process

ROS actors represent the patients within the simulation environment, which performs an animation sequence while moving around the house. These animations originate from open-source motion-capture videos and map to a hierarchial action-space taxonomy. The action-space describes specific behaviors (e.g., walking versus sitting) and any derived actions (e.g., sitting on a chair versus couch). There are virtually infinite sequences, making it challenging to record the entire universe of movement. Instead, a randomization process initializes from a recording and mutates model-joint characteristics such as flexibility, strength, and weight. This approach both increases taxonomy coverage and prevents overfitting the limited data.

#### Simulation Process

Figure 2: Simulation Instance

Diagram

Description automatically generated

ROS worlds represent the patient’s home or apartment and define models’ placement (e.g., actors and furniture), actor configuration, and devices (Bipin, 2018). Researchers use physics simulators (e.g., Gazebo) to examine interactions between these various components. For instance, the actor might perform walking to the kitchen table. Each camera will capture frames from its vantage point and transmit them to a message bus during this sequence (see Figure 2). Next, AI services subscribe to the event stream and process the visual data. Suppose the service detects a valuable signal (e.g., the refrigerator door is left open). In that case, it can post a notification to another message bus to mitigate the situation (e.g., use voice assistant).

Validating these interactions requires an ability to reconfigure these worlds without significant effort. World templating tools (e.g., AWS RoboMaker) can dynamically generate environments that meet a specification (AWS, 2021). This capability allows the researchers to create custom sensors and algorithms, not positioning furniture. That also means this dissertation aims to emphasize ROS components and world templates, not reinventing standard tooling. These components must implement an asynchronous and loosely coupled architecture.

#### Intent Extraction Process

A machine learning algorithm will process short video clips and predict the subject’s intent based on their behavior. For instance, the simulator will load a humanoid into a virtual apartment and perform a walking sequence. These animation sequences will originate from open-source databases, such as Mixamo (Adobe, 2021) and MoCap Database (CMU, 2021). IP-cameras will track the subject’s skeleton movement changes into specialized sequence-to-binary classification models. For example, one model predicts that the subject raises their hand while another assesses jumping or falling. Next, an ensemble classification algorithm combines these binary predictors into a sophisticated intent. This approach should support future researchers iteratively adding more behaviors over time.

The input sequence will contain the relative positional changes to the subject’s skeletal joints (see Figure 3). There are several potential implementations, and those solutions must perform within the hardware constraints of an edge appliance. For instance, the simulated home might produce data from dozens of cameras and sensors. Suppose the algorithm requires too many compute resources. In that case, the solution would require remote compute (e.g., public cloud), raising security and privacy concerns. Maintaining the subjects’ privacy drives specific requirements into this design, though this research defers extensive investigations to a future researcher.

Figure 3: Intent Extract Logical View



#### Rule Engine Process

Assume that the system determines that the subject has fallen, then what? Perhaps the system should ask if the person needs an ambulance through a text-to-speech device. Then, deciding which specific voice assistant adds nuances. Further complicating the matter, the fractured residential IoT market follows inconsistent protocols and standards. The second research question examines these integration challenges and proposes a rule engine. Addressing these issues requires design tenants and frameworks. While this research project explores these topics, the scope narrowly focuses on virtual devices (versus real-world integrations). These devices will likely exist as ROS plugins and services

### Feedback System Design

The second code artifact is a telemetry collection system that continuously assesses the Runtime System. Its core function is to produce the dissertation’s Results section (chapter four).

#### Decision History Store

A NoSQL time-series database records extracted intents, rule engine reactions, and various critical messages. These data points contain a foreign key to the experiment identifier and an association to the test case definition. This data store hydrates using a similar pattern for a subset of critical messages. Standard tooling already exists for recording ROS topics and persisting into binary files. Complete topic dumps will also exist outside the time-series database for troubleshooting requirements specifically.

#### Aggregation Process

Residential homes have infinite configurations and permutations with unique floor plans, furniture layouts, camera placement, noise sources, and other distinctions influencing the solution’s accuracy. Unlike a physical home, the simulator leverages ubiquitous cloud resources to scale testing across numerous virtual homes. Each simulation instance mutates its exact data through a randomization process by modifying the actors’ flexibility, weight, and other variables. The Aggregation Process is responsible for grouping these variations and calculating range statistics. Suppose the patient has fallen predictor’s accuracy could depend on the amount of furniture in the room. In that case, the results chapter will need to quantify this influence through some data pivot and summation. This research does not aim to implement a novel aggregation system and defers to industry-standard tooling (e.g., Apache Spark).

#### Evaluation Process

Creating high-quality software requires quality assurance procedures. There are several classes of defects for applications using simulation environments with AI/ML and CV, such as mixing-up actions, model non-convergence, model overfitting, code defects, performance degradation, and other issues. Automation can discover a subset of these problems using the Aggregation Process and Test Case Definitions. For example, the test case specifies that the actor will perform the jumping animation sequence. Suppose the intent prediction assumes the subject was instead sitting. In that case, the evaluation process can easily detect and report the failure. Then, specific erroneous actions and configurations require triage and troubleshooting.

#### Report Generation Process

A simple test-cases has a subject performing an animation within a world. Derived test cases could also cover entire open-source Motion Capture (MoCap) databases through scripting and templating. Next, the data generation and simulation processes will run those experiments multiple times under different world configurations. This combinatorial property creates the need for a report generation process that collects and visualizes the evaluation assessments. Building a custom Business Intelligence (BI) solution is outside this project’s scope, so this project defers to industry-standard tooling (e.g., PowerBI and Tableau). Also, budgetary limitations will prohibit exploring every combination. Instead, this research will strategically choose representative examples within the supported action space.

## Population and Sample

## Instrumentation

## Operational Definition of Variables

## Study Procedures

## Data Analysis

## Assumptions

## Limitations

## Delimitations

## Ethical Considerations

## Summary