Week 7: Refine the Design

Nate Bachmeier

TIM-7245: Directed Constructive Research

August 8, 2021

Northcentral University

# Refine the Design

## Proposed Research Methodology

Constructive design is one of the most common research methods for information systems and technology (Silvestrini & Sammito, 2012). These studies identify a problem, build solution artifacts, and communicate the implementation’s unique value (Hevner et al., 2004). In addition, many researchers follow this process to build proof-of-concept and execute case studies. Therefore, this methodology is appropriate for examining the effectiveness of the ECSOS solution and its abilities to improve elderly care.

## Artifacts

This research project has three core components which collectively form a proof-of-concept implementation and mechanism to measure results. First, the team installs WiFi-enabled Eufycam 2C cameras to collect short recordings. These cameras use motion-sensing to trigger short Audio/Video (A/V) recordings (fifteen to sixty seconds). After the filming completes, its controller (Eufy Homebase) automatically uploads the file to Network Attached Storage (NAS). The file creation event triggers an analysis workflow that extracts and publishes metadata to message buses. Developers can author extensions using Function as a Service (FaaS) constructs that subscribe to the notifications.

Second, a machine learning algorithm will classify and annotate the Human Activity Recognition metadata. There are several potential implementations (e.g., Open Pose versus Toyota’s approach). The performance and resource requirements between these strategies must exist. Ideally, the model can run in an edge appliance versus uploading into a Public Cloud Service (PCS). However, this raises concerns that the device has sufficient computing capabilities (e.g., parallel processing dozens of cameras). If analysis occurs within the cloud, then it introduces security and privacy concerns. The artificial intelligence algorithm would require additional complexity to address these risks (e.g., supporting CKKS HE encryption protocols).

Third, the ECSOS solution routes the metadata into monitoring and response extensions. These extensions include central services (e.g., identity and state management) and auditing capabilities (e.g., inputs, predictions, and recommendations). One crucial extension is the central audit logs. These tables are queryable within a NoSQL time-series database (e.g., Influx). This technology provides two essential capabilities, native support for tracking system performance across time and Schema-at-Read versus Schema-at-Write (SAR versus SAW) semantics. Datastores that support SAR are more flexible and quickly adapt to future enhancements (e.g., extending data contracts).

## Contributions

The core contribution to the body of knowledge is the case study using the proof-of-concept design. Existing research reviews each component in a silo or distinctly different use cases (e.g., sports injuries). Das et al. (2019) explain that those resources are not directly reusable, and implementations must use domain-specific labeled content. This design requirement necessitates compositing a new solution from custom and open-source software.

Second, the research produces a purpose-built machine learning algorithm made available for elderly care action recognition. This deliverable also includes quantitative metrics that describe the algorithm’s resource utilization and F-measure accuracy. Data scientists use F-measurements as a “way of combining the precision and recall of the model, and [defines] the harmonic mean of the model’s precision and recall (Wood, n.d.).” Researchers can make trade-offs in their solution to optimize this value for their specific scenario. For instance, a critical health management system might enforce higher penalties on false negatives than over positives.

## System Architecture

Elderly Care SOS requires cameras, network storage, and a custom-built appliance (see Figure 1). Optionally patients can extend the system with various CPS device integrations (e.g., remote smoke detector). The appliance must have enough computing and storage resources to perform model predictions, persist state, and execute several micro-services. Periodically, the on-premise system needs to synchronize with an external cloud component. These synchronization operations include sending status reports, downloading updates, and issuing assistance requests.

Figure 1: Abstract Design

Diagram

Description automatically generated

## User Roles

Aside from patients, there are three additional user roles: nurses and healthcare providers, family members, and administrators. These users can use a mobile app or web portal to access the relevant data. All operations from either UI (User Interface) require Authentication, Authorization, and Auditing (AAA). When systems mandate AAA enforcement, it prevents negligence or malicious actions while increasing transparency. It is also critical that the patient maintains control of their privacy. For example, they might want to share a weekly aggregate health report with family members, not verbose details.

## System Reliability

The architecture’s components communicate over WiFi, Zigbee, and Bluetooth protocols. These messages are likely to encounter transfer failures due to radio interference or devices being offline. There must be support within the message buses to cache and reattempt any message delivery failures using exponential backoff policies. Otherwise, the state management’s perspective can become distorted. Time-sensitive messages (e.g., the subject has fallen) require a primary and secondary communication channel, such as phone line or mobile phone pairing.

The appliance must locally run several services that handle core scenarios like identity and message routing. Developers can also load custom extensions that subscribe to event notifications. Those various subsystems require isolation and controls to limit the blast radius of a specific failure. An industry-standard approach would be to use micro-service designs and container orchestration technologies (e.g., Kubernetes)(Wen et al., 2020). These products can manage fail-over replicas and promptly restart crashed instances.

## Data Collection Process

Most information enters the system through the WiFi cameras. Ideally, those cameras are accessible only through a dedicated Virtual Local Area Network (VLAN). This recommendation protects the unencrypted Real-Time Streaming Protocol (RTSP) from eavesdropping and tampering attacks. After the video clip is available, ECSOS must process it through several machine learning models (e.g., facial recognition, object detection, and action recognition). These metadata annotations persist into a time-series database. Lastly, populating the database requires the patient(s) to behave normally and let the system collect the video recordings.

Initially, the system will not have any training data and cannot make recommendations. However, researchers can accelerate data labeling with online products like Amazon SageMaker Ground Truth (Amazon, 2021). This service offers clustering capabilities to group related artifacts and streamlines manual tasks. Alternatively, users can crowd-source labeling jobs through Amazon Mechanical Turk.

## System Validity

There must be feedback loops that confirm that the predictions and recommendations are accurate. Without this capability, it would be challenging to discover issues and prioritize machine learning model changes. One potential solution is to collect these responses through a patient mobile app. App users can enumerate previous recordings and see the associated metadata. Those filmings are subject to a retention policy that automatically deletes old content. If they disagree with the predictions, they can make corrections inline. After making the manual update, the user can help improve the experience by submitting the footage to the ECSOS Cloud.

The ECSOS Cloud maintains a history of all incorrect predictions. Data scientists can review those responses, look for patterns (e.g., mixing up two actions), and make the necessary modifications. There must be some mechanism to include user feedback and avoid introducing biases into the model (García-Pérez, 2012). For example, an individual user could post thousands of feedback comments containing inaccurate data change requests. This situation could regress other user’s experiences. Similar biases can enter the system due to insufficient test subject’s racial diversity (e.g., only validating white men).

## Quotas and Limits

The appliance and other components have physical capacity constraints. For example, the computing resources might support real-time analysis across sixteen camera sources. If the household wants to provision thirty-two cameras, then they need to install a second appliance. Systems that declare their quotas and limits can ensure that the user’s configuration works and provide a positive user experience. During the case study, the researchers will determine reasonable appliance support limits. These limits must minimize the hardware costs proportional to the monitored environment’s total cost.

After selecting limits, the engineering team must validate that the system achieves Service Level Objectives (SLO). Next, operations teams assess SLO attainment through a Quality of Service (QoS) model, which considers availability, reliability, response time, and throughput. Finally, the service administrators require a mechanism to centralizing this telemetry into the ECSOS Cloud. This capability enables the service team to uncover issues at remote patient homes.