BUSINESS PLAN

Applicant

Yong-Jun Shin (yshin1209@gmail.com) CV

- Assistant Professor of Biomedical Engineering at the University of Connecticut
- Received M.D. and Ph.D. in electrical engineering
- Received the National Science Foundation Smart and Connected Health Award
- Received the Microsoft Azure Research Award

Purpose

Democratize medical knowledge by providing patient-oriented clinical decision support.

Problem

Patients are often asked to be involved in clinical decision making. In that respect, clinical decision support can be helpful not only to healthcare practitioners but also to patients. Patient-oriented clinical decision support may also enable patients (or their families) to detect important clinical findings that busy practitioners might miss. As individual health records are becoming increasingly available on mobile devices (e.g., Apple Health Records), user-friendly and intuitive clinical decision support geared towards patients can be immensely useful to realize true patient-centered personalized healthcare. Furthermore, the history of clinical decision support stored in a personal storage may legally protect patients' rights.

Many computer-based clinical decision support approaches have been developed over the past 50 years. Two widely-used approaches are 1) machine-learning and 2) logic-based reasoning. In contrast to machine learning, logic-based reasoning captures medical knowledge that has been accumulated and refined by human intelligence for centuries, in the form of medical textbooks and clinical guidelines. Although there have been many examples that demonstrated its value, it is surprising that logic-based reasoning has not led to its broad adoption. Deploying simple forms of logic-based reasoning such as an "if … then" logic (e.g., IF blood sodium is less than 135 mEq/L THEN the diagnosis is hyponatremia) in a single computer is straightforward. The real challenges arise not from the single use of a simple logic but from (1) how the logic can interoperate with other logics to form a complex, interconnected reasoning process, (2) how such reasoning process can readily adapt to changes in underlying medical knowledge, (3) how it can be event-driven while allowing human intervention when needed, (4) how it can be interoperable (i.e., language and platform independent), and (5) how it can be widely available and scalable.

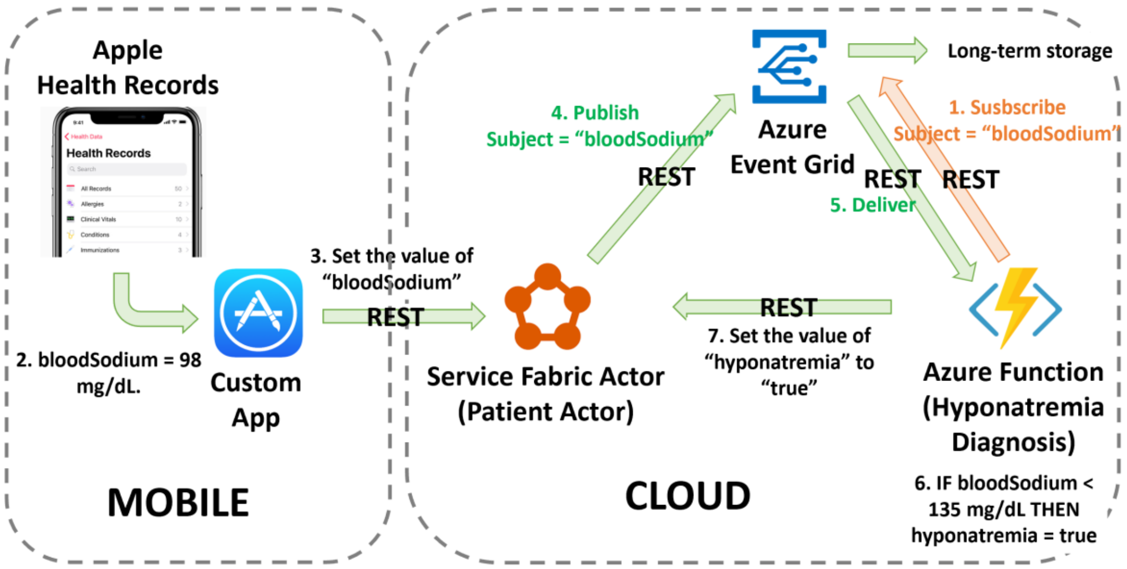
Solution

Clinical decision support can be considered as a part of the smart and connected healthcare system. Since healthcare is complex and may change over time, the variables (other names: properties, fields) and functions (other names: methods, actions) of each system component cannot be clearly defined in advance. In addition, the interactions among the components triggered by events may dynamically change. Therefore, hard-coding or tight-coupling the variables, functions, and event-handling of individual system components becomes cumbersome in the presence of such uncertainty. Although a microservice framework enables independent scalability and increased flexibility compared to traditional monolithic approaches, variables, functions, and event-handling are still tightly-coupled within each microservice. In this context, we propose **"Serverless Nanoservices"** as a new computing paradigm for smart and connected healthcare:

1. **Variable Nanoservices**: Azure Service Fabric provides microservice-enabled reliable actors which are isolated, independent units of compute and storage. Each actor is assigned a unique ID and can represent an individual patient. REST-enabled Variable Nanoservices can be developed and consumed to dynamically add, update, or remove actor variables with no downtime.
2. **Function Nanoservices:** Azure Functions, AWS Lambda**,** Google Cloud Functions, and IBM Cloud Functions are the examples of compute services that can provide REST-enabled Function Nanoservices. Clinical decision support logics can be represented by individual Functions which are reusable and independently scalable.
3. **Event-handling Nanoservices:** Azure Event Grid**,** Amazon Simple Notification Service, Google Cloud Pub/Sub are the examples of fully-managed event-routing services that can provide proposed REST-enabled Event-handling Nanoservices.

Simple Proof-of-Concept Demo (Hyponatremia Diagnosis)

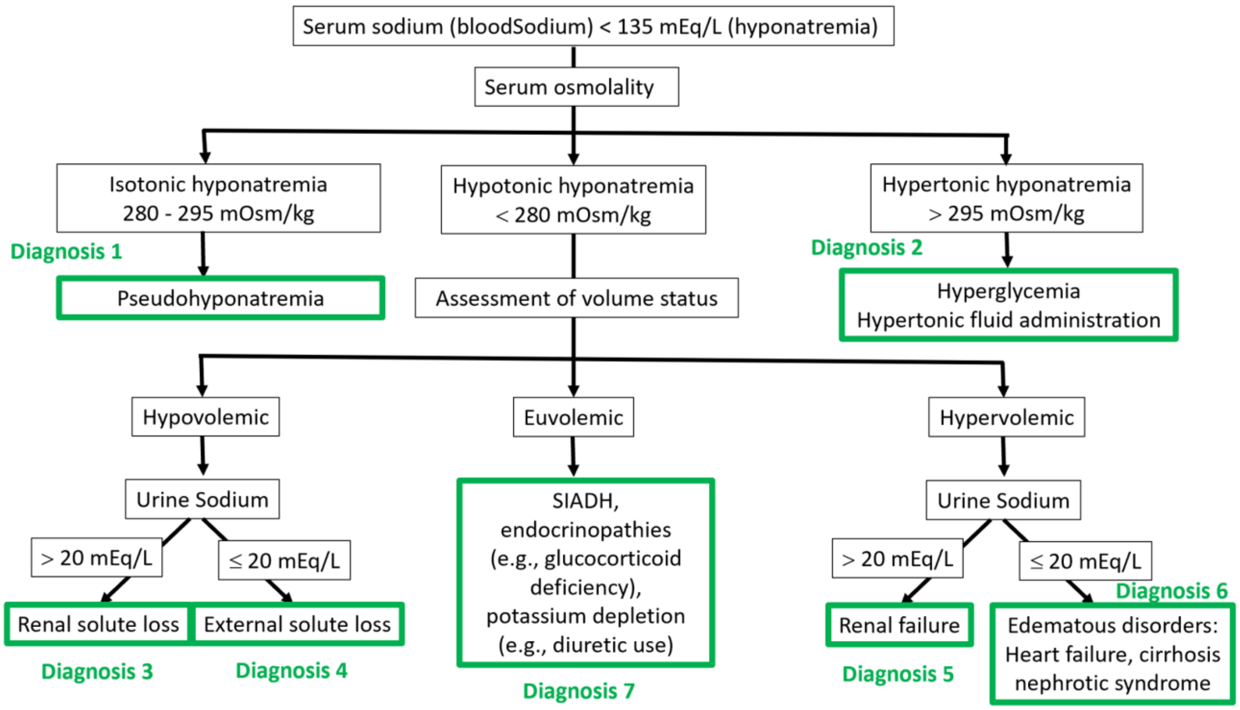
Hyponatremia is a condition that occurs when the level of sodium in the blood is abnormally low (less than 135 mEq/L). This demo assumes receiving a blood sodium value from Apple Health Records and determines whether there is hyponatremia or not using the proposed Serverless Nanoservices approach. For this demo, Azure Service Fabric Reliable Actors, Azure Functions, Azure Event Grid, and Visual Studio are used for increased productivity and seamless integration.


Apple Health Records
Health Records
All Records
Allergies
Clinical Vitals
Conditions
Immunizations
2. bloodSodium = 98 mg/dL.
Custom App
MOBILE
3. Set the value of "bloodSodium"
REST
Service Fabric Actor (Patient Actor)
4. Publish Subject = "bloodSodium"
REST
Azure Event Grid
Long-term storage
1. Susbscribe Subject = "bloodSodium"
REST
REST
5. Deliver
REST
7. Set the value of "hyponatremia" to "true"
Azure Function (Hyponatremia Diagnosis)
6. IF bloodSodium < 135 mg/dL THEN hyponatremia = true
CLOUD

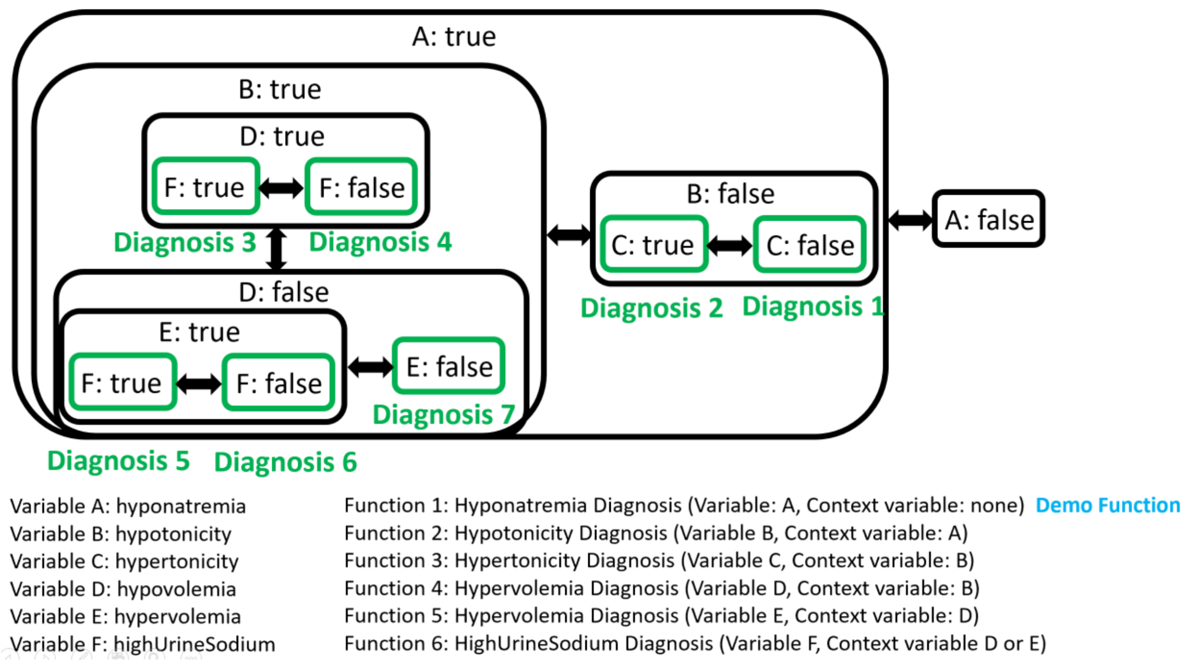
Only the cloud part is demonstrated in the demo and please download NanoServiceAPI Demo.pdf for more information.

Context-aware Interconnected Functions (Logics)

When events occur, the sequence of the events matters. Therefore, it is important to handle any event in a specific context which reflects the history of previous events. This can be illustrated using the differential diagnosis of hyponatremia. A rapid decrease in serum sodium levels, i.e., hyponatremia, over 24 to 48 hours can lead to severe cerebral edema and the central nervous system symptoms which is an acute medical emergency. Without an appropriate intervention to increase sodium level, patients can develop coma, brainstem herniation, and respiratory arrest that may lead to death. Therefore, finding out the cause of hyponatremia is critically important to treat the patient and a flowchart can be used to guide differential reasoning as shown below**.** Note the reasoning process may involve human intervention (e.g., assessment of volume status) which can be done using serverless notification services such as Azure Notification Hubs.

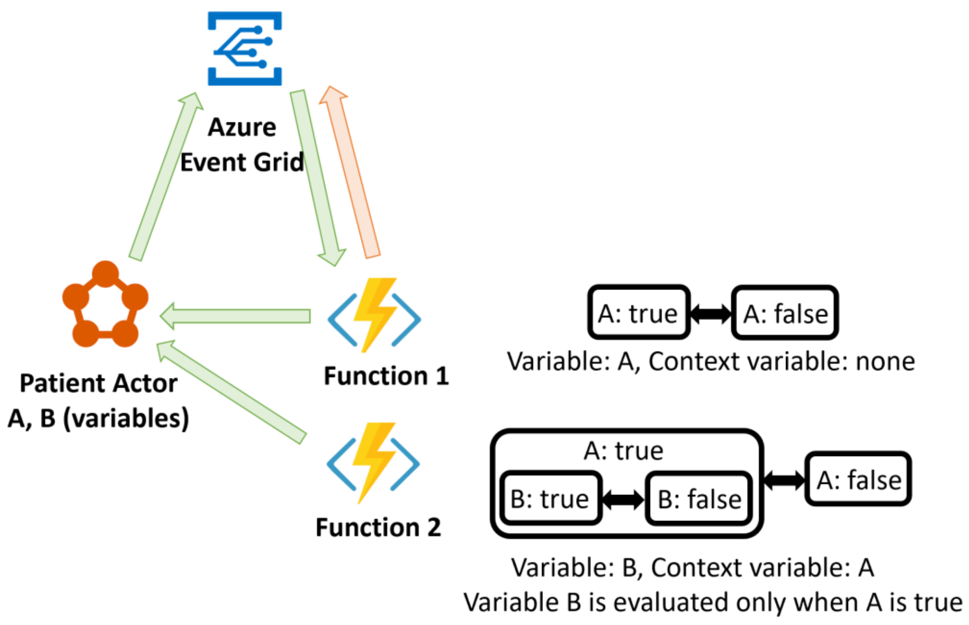

Serum sodium (bloodSodium) < 135 mEq/L (hyponatremia)
Serum osmolality
Isotonic hyponatremia 280 - 295 mOsm/kg
Hypotonic hyponatremia < 280 mOsm/kg
Hypertonic hyponatremia > 295 mOsm/kg
Diagnosis 1
Pseudohyponatremia
Assessment of volume status
Diagnosis 2
Hyperglycemia Hypertonic fluid administration
Hypovolemic
Euvolemic
Hypervolemic
Urine Sodium
> 20 mEq/L
≤ 20 mEq/L
Renal solute loss
External solute loss
Diagnosis 3
Diagnosis 4
SIADH, endocrinopathies (e.g., glucocorticoid deficiency), potassium depletion (e.g., diuretic use)
Diagnosis 7
Urine Sodium
> 20 mEq/L
≤ 20 mEq/L
Diagnosis 6
Renal failure
Diagnosis 5
Edematous disorders: Heart failure, cirrhosis nephrotic syndrome

Although flowcharts are useful, their flat structure can easily become messy because it cannot capture the hierarchical, modular, concurrent features of complex and dynamic differential reasoning. In contrast, those features can be well represented by a statechart as shown below. The whole process is decomposed into six loosely-coupled serverless Functions which are independently scalable and reusable. For example, Function 6 (Variable F) is reused in the context of Variable D and Variable E. Also note that each Function needs to care only its "immediate context", thanks to the hierarchical structure. For instance, Function 6 needs to check only context variable D because if D is active (either true or false) it automatically means B (higher level context variable) is true and A (highest level context variable) is also true. In this



Variable A: hyponatremia	Function 1: Hyponatremia Diagnosis (Variable: A, Context variable: none) Demo Function
Variable B: hypotonicity	Function 2: Hypotonicity Diagnosis (Variable B, Context variable: A)
Variable C: hypertonicity	Function 3: Hypertonicity Diagnosis (Variable C, Context variable: B)
Variable D: hypovolemia	Function 4: Hypervolemia Diagnosis (Variable D, Context variable: B)
Variable E: hypervolemia	Function 5: Hypervolemia Diagnosis (Variable E, Context variable: D)
Variable F: highUrineSodium	Function 6: HighUrineSodium Diagnosis (Variable F, Context variable D or E)

fashion, the interconnection of clinical decision support logics (Functions) is greatly simplified since each logic needs to interact with only its "immediate context" logic(s). This point is illustrated more clearly using two Functions below. Function 1 determines the value of A while Function 2 determines the value of B. Since the context variable for Function 2 is "A" which is determined by Function 1, Function 2 is executed only when A is true.


Azure
Event Grid
Patient Actor
A, B (variables)
Function 1
Function 2
A: true
A: false
Variable: A, Context variable: none
A: true
B: true
B: false
A: false
Variable: B, Context variable: A
Variable B is evaluated only when A is true

Target Lab Tests and Clinical Decision Support
The differential diagnosis of hyponatremia may not be the most common usage although it shows how our approach works. We will initially target common lab tests (blood and urine tests) and clinical decision support that collectively makes use of the lab data for differential reasoning.

Why Now?

- Apple opened Health Records API to developers in June 2018 [link].
- A comprehensive serverless computing infrastructure is becoming available recently:
  - Azure Event Grid became generally available in January 2018 [link].
  - Azure Functions became generally available in November 2016 [link].
  - Azure Service Fabric became generally available in March 2016 [link].

Market Potential

Institutions that support Apple Health Records on iPhone include Johns Hopkins, NYU Langone Health, Penn Medicine, UC San Diego Health, Weill Cornell Medicine, and Yale New Heaven Health. The list can be found here which is over 50 now and keeps growing. It is likely that similar health records will become available to Android users in coming days.

Competition / Alternatives

- There have been efforts to integrate clinical decision support with electronic health records for healthcare practitioners (e.g., CDS Hooks). However, due to the reasons described in the "Problem" section, logic-based clinical decision support has not been widely adopted. In this regard, our Serverless Nanoservices approach can be applied to clinical decision support for health professionals as well.
- There are self-checking online tools for patients (e.g., Mayo Clinic Symptom Checker) but these are simple guides and do not provide clinical decision support.

Business Model

- Patients: Subscription
- Developers: API monetization (they can develop iPhone apps that consume our Serverless Nanoservices)
- Healthcare practitioners: API monetization (they can develop Functions for differential reasoning and provide them as APIs)

Team

Yong-Jun Shin, MD PhD CV
I am the director of the Computational and Systems Medicine Lab at the University of Connecticut. I'm interested in applying digital logic (statechart) and feedback control / estimation theory to systems medicine. My medical education taught me how complex medicine can be and through

my engineering education (MS and PhD in electrical engineering) I learned computational approaches to design, analyze, and control complex and dynamic systems. In this regard, my research interest has been applying the principles and insights that I learned in engineering to biological systems (e.g., the human body) which are also complex and dynamic. I've been also deeply intrigued by emerging disruptive technologies such as serverless computing, Internet of Things (IoT), and augmented reality which will fundamentally change science and engineering. I enjoy coding and the demo shown in this plan is my work.

Luis A. Serrano
Luis is a Computer Science and Engineering PhD student who has been working on advanced applications of the Serverless Nanoservices approach described in this plan. For more information, please visit https://csml.uconn.edu/ .

Financials

I'm currently working as an assistant professor and the 2-year National Science Foundation grant titled "Distributed and adaptive personalized medicine" awarded in August 2017 (amount: $288,056) has been supporting my PhD student Luis and providing cloud resources needed.

Vision

Medical knowledge is best understood by healthcare professionals. Therefore, we plan to develop a user-friendly interface that they can use to easily transform their knowledge into Serverless Functions. Serverless Functions can provide not only medical knowledge but also mathematical algorithms for physical model-based tasks (e.g., controlling blood glucose levels so that they stay between 70 and 180 mg/dL). Two such mathematical algorithms (PID control as a service and LMS adaptive parameter estimation as a service) can be found at https://github.com/uconn-csml/EaaS. Integrating Azure IoT Hub and Azure IoT Edge with the proposed Serverless Nanoservices, we will eventually provide a platform where practitioners, researchers, and engineers can readily develop serverless cloud / IoT / edge applications for healthcare (one example is "Distributed and adaptive personalized artificial pancreas" and for more information please visit https://csml.uconn.edu/).