

Design and implementation of the temperature component in Personal Healthcare Slow Intelligence System

Zichuan “Jerry” Ye

Department of Computer Science, University of Pittsburgh

System Description:

The Personal HealthCare Slow Intelligence System is a system that collects, processes healthcare data and delivers messages among multiple functional components. This system is built upon a component-based architecture, in which each component is responsible for a certain source of healthcare data or other functions such as uploading and graphic user interface. The components communicated with the central server “SIS-server” by sending messages. Each type of message among components is numbered and well-defined. By having access to the number of the messages that are carried by SIS-server, the components recognize the ones that are needed and fetch them from the server. This message-oriented communication supports the flexibility and scalability of the system. It also helps keeping the integrity of the system and expedites the creation, management and termination of the components.

In this report, a new implementation of a component that is aimed to handle the temperature data is proposed. A component “Temperature” is described and implemented. Fundamental design is shown in Figure 1. It is noted that communications among all components are message-based and can only be carried by SIS Server.

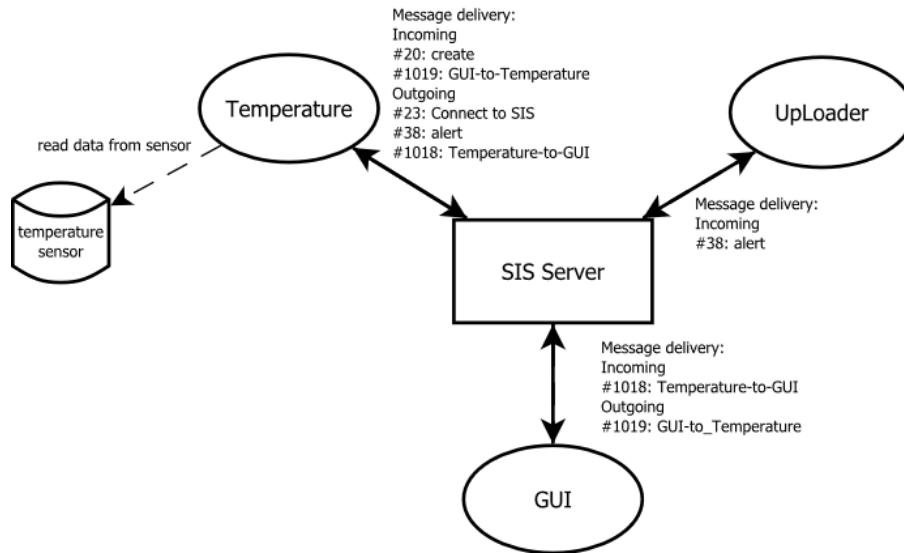


Figure 1. A component-based architecture that consists a component for temperature. Multiple functional components communicate via SIS-messages that are delivered by SIS Server. For simplicity reason, the other components such as Blood-Pressure, SPO2 and ECG in SIS Healthcare are now shown here.

The fundamental work flow of the Temperature component can be described into the following steps and depicted in Figure 2.

1. The temperature component is coded and created from as an individual component (CreateTemp.java). It starts an executable program (TEMPerV21.exe) when begins to run. It needs to be mentioned that “TEMPerV21.exe” is compiled runnable that handles I/O of the temperature sensor and writes temperature data to a local file. By default, the data files is located at “D:\Temp”. Due to the limit of software/hardware resources, we must use this executable to receive data from a temperature sensor and store them on local disk.
2. A user set up parameters of the temperature component, including sample refresh rate, sensor start time, sensor end time, temperatures to trigger high/low temperature alerts.

3. After GUI completed reading the user input, the parameters are packed into an SIS-message (#1019) and sent to SIS-server, which later delivers the message to Temperature Component.
4. The temperature component receives the message and decode it. It then starts processing the data according to the parameters that are carried in the message. During its execution, outgoing messages are created and sent out in two scenarios: (a) the temperature component reports to the GUI via messages (#1018) periodically to for current temperature readings, and (b) when the temperature triggers some threshold (high or low temperature), it sends a messages (#38) to the UpLoader.

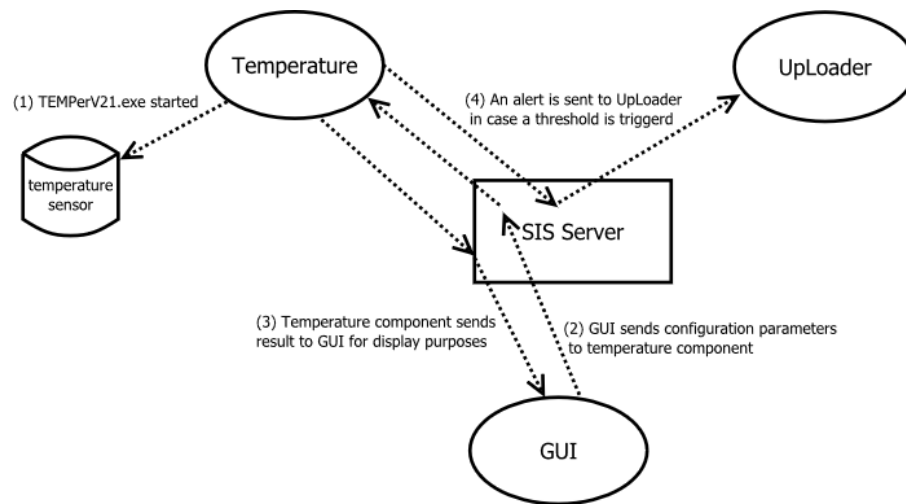


Figure 2. A brief flow diagram describes how temperature component works and communicated with other components.

In the programming point of view, the codes for temperature component, GUI and uploader are located in three different directories and have NO internal connection/call to each other. As parts of previous development, GUI (CreateGUI.java) and uploader (CreateUpLoader.java) components were written already. To expedite setting parameters of the temperature component, a class “TempSettings.java” is implemented to create a

pop-out window for the main GUI panel and to record user inputs regarding to temperature settings.

Messages:

The content of each message that is involved in the communication among Temperature component, GUI and UpLoader is defined according:

For more detailed format and content for each message, please see at the appendix.

Working Demo:

Will be presented on 11/4/2014.

Appendix:

Message #20: Component Creation

```
<?xml version="1.0" standalone="yes"?>
<!--Generated by SISProjectCreator Version 1.0-->
<Msg>
  <Head>
    <MsgID>20</MsgID>
    <Description>Create Temperature Component</Description>
  </Head>
  <Body>
    <Item><Key>Name</Key><Value>Temperature</Value></Item>
    <Item><Key>Password</Key><Value>****</Value></Item>
    <Item><Key>SecurityLevel</Key><Value>3</Value></Item>
    <Item><Key>SourceCode</Key><Value>CreateTemp.jar</Value></Item>
    <Item><Key>InputMsgID1</Key><Value>1019</Value></Item>
    <Item><Key>OutputMsgID1</Key><Value>1018</Value></Item>
    <Item><Key>OutputMsgID2</Key><Value>38</Value></Item>
    <Item><Key>Component Description</Key><Value>This component initiates and processes the
temperature data.</Value></Item>
  </Body>
</Msg>
```

Message #23: Connection to SIS Server

```
<?xml version="1.0" standalone="yes"?>
<!--Generated by SISProjectCreator Version 1.0-->
<Msg>
  <Head>
    <MsgID>23</MsgID>
    <Description>Connect to SISServer</Description>
  </Head>
  <Body>
    <Item><Key>Name</Key><Value>Temperature</Value></Item>
  </Body>
</Msg>
```

Message #38: General Alert Message

```
<?xml version="1.0" standalone="yes"?>
<!--Generated by SISProjectCreator Version 1.0-->
<Msg>
  <Head>
    <MsgID>38</MsgID>
    <Description>Alert Message</Description>
  </Head>
  <Body>
    <Item> <Key>Alert_Context</Key> <Value>alert_context</Value> </Item>
    <Item> <Key>Date</Key> <Value>alert_date</Value> </Item>
    <Item> <Key>FName</Key> <Value>patient_first_name</Value> </Item>
    <Item> <Key>LName</Key> <Value>patient_last_name</Value> </Item>
    <Item> <Key>send_all</Key> <Value>True/False</Value> </Item>
    <Item> <Key>dr_email</Key> <Value>email_of_doctor</Value> </Item>
    <Item> <Key>hospital_email</Key> <Value>email_of_hospital</Value> </Item>
  </Body>
</Msg>
```

Message #1018: Temperature _to _GUI

```
<?xml version="1.0" standalone="yes"?>
<!--Generated by SISProjectCreator Version 1.0-->
<Msg>
  <Head>
    <MsgID>1018</MsgID>
    <Description>For GUI to display temperature</Description>
  </Head>
  <Body>
    <Item> <Key>Temp_enable</Key> <Value>True/False</Value> </Item>
    <Item> <Key>Date</Key> <Value>sample_date</Value> </Item>
    <Item> <Key>Temp</Key> <Value>sample_temperature</Value> </Item>
  </Body>
</Msg>
```

Message #1019: GUI_to_Temperature

```
<?xml version="1.0" standalone="yes"?>
<!--Generated by SISProjectCreator Version 1.0-->
<Msg>
  <Head>
    <MsgID>1019</MsgID>
    <Description>For Temperature Component setting up</Description>
  </Head>
  <Body>
    <Item><Key>Temp_enable</Key><Value>True/False</Value></Item>
    <Item><Key>Refresh_rate</Key><Value>sample_date</Value></Item>
    <Item><Key>Start_time</Key><Value>component_start_time</Value></Item>
    <Item><Key>End_time</Key><Value>component_end_time</Value></Item>
    <Item><Key>High_temp_alert</Key><Value>high_temperature_for_alert</Value></Item>
    <Item><Key>Low_temp_alert</Key><Value>low_temperature_for_alert</Value></Item>
    <Item><Key>send_all</Key><Value>True/False</Value></Item>
    <Item><Key>dr_email</Key><Value>email_of_doctor</Value></Item>
    <Item><Key>hospital_email</Key><Value>email_of_hospital</Value></Item>
  </Body>
</Msg>
```