

Chemistry By Mobile Phone (or how to justify more time at the bar)

Jamie M, Robinson¹; Jeremy G, Frey¹;

Andy J, Stanford-Clark²; Andrew D, Reynolds²; Bharat V, Bedi²;

1. School of Chemistry, University of Southampton, SO17 1BJ, United Kingdom

2. IBM UK Laboratories, Hursley Park, SO21 2JN, United Kingdom

Email Contact : j.m.robinson@soton.ac.uk

Abstract

By combining automatic environment monitoring with Java smartphones a system has been produced for the real-time monitoring of experiments whilst away from the lab. Changes in the laboratory environment are encapsulated as simple XML messages, which are published using an MQTT compliant broker. Clients subscribe to the MQTT stream, and produce a user display. An MQTT client written for the Java MIDP platform, can be run on a smartphone with a GPRS Internet connection, freeing us from the constraints of the lab. We present an overview of the technologies used, and how these are helping chemists make the best use of their time.

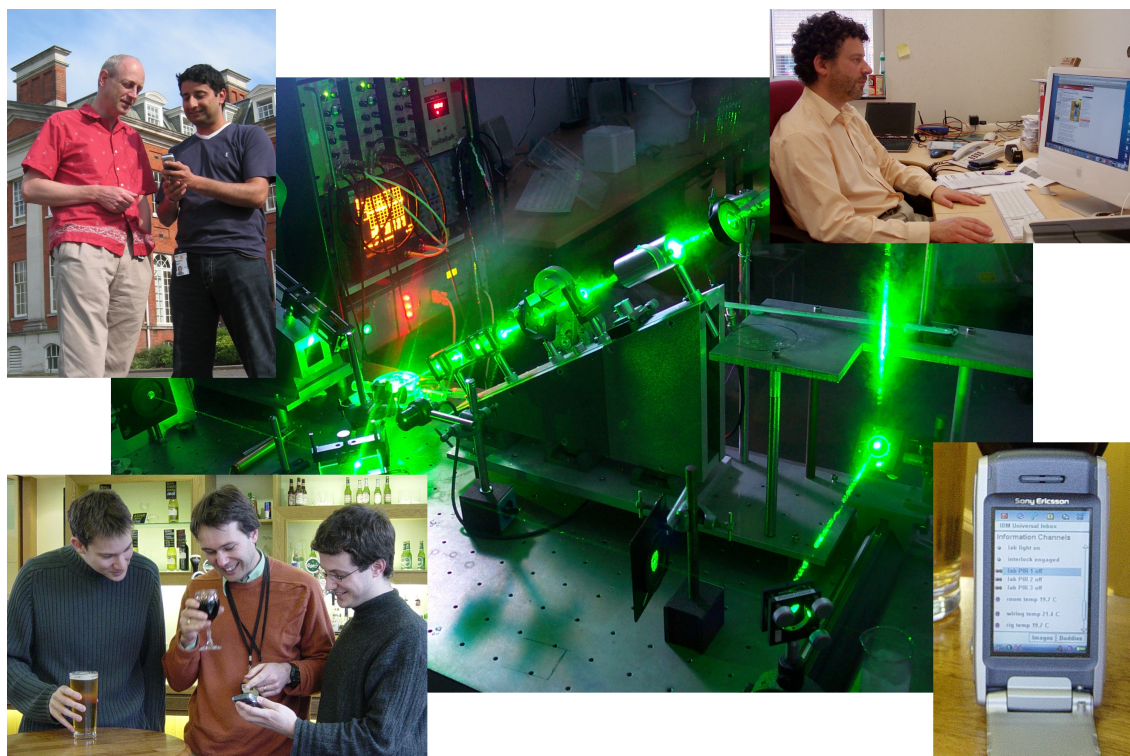


Illustration 1: Our laboratory can be monitored remotely, and simultaneously by the Scientist, their Supervisor, and Industry Collaborators

1. The Chemistry Experimental Problem.

Improvements in automation technology have made it increasingly common to leave practical experiments in Chemistry running unattended. In many cases this can lead to a safer working environment (e.g. experiments involving ionising radiation or laser sources) and better results (for example, liquid surface experiments can be sensitive to vibration).

However, by being present during an experiment, an experienced chemist will notice problems as they occur and either alleviate them, or abort the experiment early and restart it, having taking measures to avoid the problem recurring. Hence we need to provide the chemist with the ability to monitor their experiment and its environment whilst taking advantage of the safety and quality of result improvements that can be gained by allowing an experiment to run unattended.

Using a combination of off the shelf electronic components, and a software solution from IBM UK laboratories, a system has been created that allows for the real-time monitoring of the chemistry laboratory from a variety of clients.

By using standards-compliant middleware, a system was implemented that can be easily extended to include other sensors and output devices. The use of a “data broker” allows for the addition of new devices, without any need to adjust the rest of the system.

In this work we present the laboratory information feed using a Java dashboard interface running on smartphone. This interface represents the information streams from the lab using graphical icons, with additional detailed notes available. The dashboard is also capable of generating audio or motion alerts (depending on the phone's capabilities) indicating to the user when an error condition has occurred, or to prompt a user response. It is planned to extend the system to allow the remote user to provide the laboratory with feedback, which could be used to drive an analysis procedure, or to modify experimental conditions.

2. Background.

1 Middleware

Middleware is the name given to software which provides a messaging fabric to link applications and systems together. The implication from the name is that it is something that occupies the space between the operating system and the applications, and this is pretty

much accurate. The alternative to not using a middleware system, is that the application writer has to deal with the mechanics of getting messages from A to B, dealing with connection failures, network outages, duplicate messages, etc, etc.

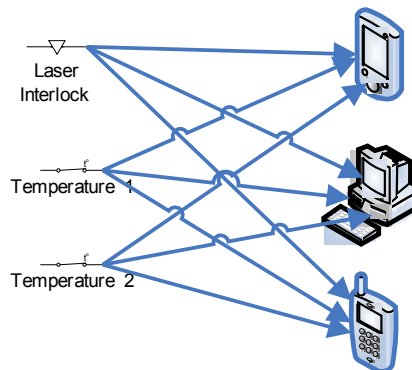


Illustration 2: Client Server DataFlow without Messaging Middleware. All producers talk directly to the clients, and have to deal with networking instabilities.

A middleware system takes that responsibility away from the application writer, and provides a convenient interface to enable the application to send a message, and be confident that it will get to its destination. This enables the application writer to focus on the domain-specific part of the problem (i.e. closer to the user and the data), and not have to worry about moving messages around the system. So with a middleware system, the problem of “by what mechanism will a message be delivered from application A to application B”, becomes: “what would you like to send, and what will you do with it when it arrives?”

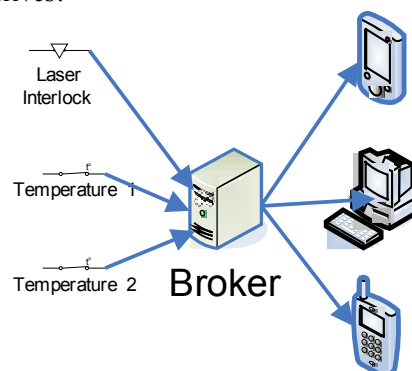


Illustration 3: Data Flow with Messaging Middleware. All data flows through the broker, Decoupling data producers from data users.

2 Using middleware in the laboratory space

A powerful feature of IBM's WebSphere MQ messaging middleware, is that it uses an

architecture which allows collaborating applications to intercommunicate via a central hub, known as a Message Broker. Each application sends its data to the broker, and the broker sends it on to the intended recipients.

This decoupling of producer (or publisher) from consumer (or subscriber) is extremely powerful, as it means that neither the publisher nor the subscriber needs to know about the other party. This means that data producers can be simply set up to publish their data to the broker, and that's all they need to do. On the other side, subscriber applications then tell the broker what kind of information they're interested in, and the broker forwards any messages that come in from publishers, matching those interests, to the interested subscribers. This permits great flexibility in the rapid exploration of new ideas and the easy deployment of new applications which are written as new uses are found for the data that's being published. Similarly, if a piece of equipment is swapped for another type of machine, as long as it publishes the same information as the previous one, none of the subscribing applications need to know that the swap has taken place: they simply continue to receive the data they expect, as before.

In an environment where things are often changing, and new things are being tried out, with the research lab being a case in point, the decoupling of publishers from subscribers has particular benefit as the back end applications which process, display, etc, the information can remain the same while the equipment generating that data may be changed, improved, exchanged, etc. The other big benefit is the one-to-many capability of publish/subscribe - several people and applications may receive one piece of data being published from a device in the lab.

3 MQTT

MQTT (MQ Telemetry Transport) is one of the protocols supported by the IBM Message Broker products as a way of getting data in and out of the broker. The protocol was designed specifically for remote telemetry applications, with three specific design goals:

- (1) It should offer a once-and-once-only assured delivery mode to enable a message to be reliably transferred all the way from a remote sensor to a back-end application (for example the number of gallons of oil delivered to a customer is measured by a flow meter, and back in the Enterprise a bill has to

be generated for the customer who received the oil)

- (2) The protocol should be as lightweight as possible across the "wire" (or other communication medium): most remote telemetry is done over low bandwidth, high cost networks, and so minimising the overhead of each message is highly desirable.
- (3) The protocol should be very easy to implement on embedded devices such as sensors and gateways.

The MQTT protocol has an open, published specification, which is available for anyone to implement on a client device, and reference implementations are available free of charge from IBM in Java and C. The MQTT specification, documentation and sample code, is linked to from <http://mqtt.org>. MQTT has been supported in IBM's products for over 4 years, and is gaining a wide acceptance in many industry areas.

4 IBM Message Brokers

IBM has several "message broker" products in the product portfolio. In decreasing order of size, range of functions, sophistication and price, these are WebSphere Business Integration Message Broker (WBIMB), WebSphere Business Integration Event Broker (WBIEB), and WebSphere Connection Server Micro Edition (WCSME). The first, WBIMB, is often referred to as the Enterprise Broker, and that runs on a server-class machine, supports multiple input and output messaging protocols, and has a rich set of tools and functions to develop message transformations and routing logic to enable the broker to act as a powerful, central communications and transformation hub for an Enterprise-scale middleware operation. By contrast the last mentioned, WCSME, more widely known as the "MicroBroker", is a small footprint (about 500K of Java) message broker, designed for use in embedded applications, such as for integrating sensors, actuators and applications at a remote location (for example, in the research lab). The MicroBroker uses MQTT as its communications protocol, and is able to selectively "bridge" some of the information topics to another broker (often an Enterprise Broker). This hierarchy of brokers implements the middleware fabric that enables the seamless delivery of messages from sensors to back-end applications, with aggregation and additional processing being performed at the point in the network where it makes most sense.

5 Publish and Subscribe Technologies

The publish/subscribe capability implemented in the IBM Message Brokers enables a one-to-many distribution of data from data producers (publishers) to data consumers (subscribers). Every message is published with a descriptive "topic", which (rather like the subject line of an email message) says what the message is "about". The topic is organised hierarchically, like a URL, so each piece of information can be slotted into the correct place in an information hierarchy or ontology.

Subscribers tell the broker what topics they are interested in, by specifying a list of topics, potentially using wild-cards, to indicate an interest in explicitly named topics, or sub-trees of the topic (information) space.

Publishers and subscribers connect to the broker using the MQTT protocol, or one of the other protocols supported by the message broker.

6 Message Push

When a message from a publisher arrives at the broker, the broker examines the topic, and matches it against the expressed interests of the currently registered subscribers (including wild-card matches). The broker then sends a copy of the message to each subscriber whose subscription matches that of the incoming publication. This is a true "push" model: data is sent from the publisher to the broker, then directly sent by the broker to the subscriber. The subscriber maintains an open socket connection (over which the MQTT protocol flows) to the broker in order to receive those pushed messages.

3. Implementation

1. Implementation Overview

Initially it was decided to monitor temperature, presence of people in the lab (and their movement in and out) and the state of the lab lights (on/off). A sensor connection to the lab laser interlock was also made, as this provides a quick indication of unauthorised access to the lab (which would result in the interlock changing state).

Data is measured by a range of sensors in the laboratory, these values are captured into a computer system, which looks for changes in the measurements. When a change is detected, a message is published as a MQTT message over an IBM MicroBroker running locally in the Southampton. These messages are then

distributed to a range of clients, some of these being end user displays, some being storage agents (for example writing data to an SQL database), and some being transform agents to reformat the data for other clients.

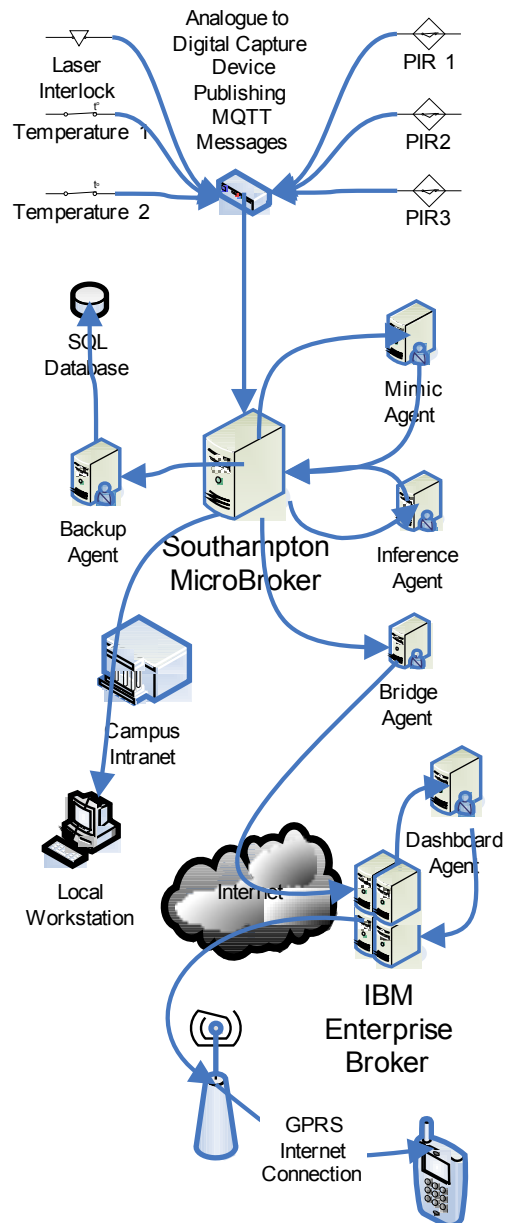


Illustration 4: Data Flows within the laboratory messaging system

2. Electronics

Temperature was measured using three semiconductor temperature sensors supplied by RS-Components (Stock No 317-960). These are supplied mounted in a TO-92 casing and provide a linear temperature to voltage output. This signal is then captured using an existing

Data Acquisition card (National Instruments LAB PC+). Currently these sensors measure temperature in three areas of the lab, away from the rig, near the rig and inside the safety covers alongside the input-side optics.

The state of room lighting (on/off) is being monitored by a photo-diode placed alongside one of the light fittings. The signal from this is monitored by the data acquisition card, and passed through a threshold filter to generate a toggle that mimics the light switch. This option was chosen in preference to a direct connection to the laboratory wiring on grounds of safety. It has the downside of monitoring only one light-fitting. A better solution may be to get a higher sensitivity photo-diode and mount this alongside the photo multiplier tube (Optical Output Sensor) on the apparatus. Currently we're only interested in light being on or off, as the actual level is reasonably constant in the region being studied however for an experiment detecting in the visible light region, it may be more appropriate to monitor the actual light level.

Monitoring of personnel movements was performed using security alarm components. A PIR sensor in the lab detects people moving, it has been found that due to the layout of the lab, and the necessity of operators to stay still, that the PIR loses people in the lab, and stops triggering. This was predicted when we were at the planning stage, so the ability to check people passing through the lab doors was also added. Door state is monitored using simple magnetically controlled reed switches. All these toggle sensors connect back to the TTL Input of the Data Acquisition Card, using the card's internal 5V supply with a bias resistor to generate the high state.

3. Transformation Agents

Mimic Agent

The workstation client shown in the overview illustration 4 includes an applet that plots live data from the broker as it is received. This plotting applet is linked to the temperature data feed. However the sensor system in the lab only produces messages when the temperature changes. Hence in the default state, only these changes are plotted, with no reference to time. A more useful plot is one showing temperature over the most recent time period, to get the applet used to plot this required publishing a message to it every second. To achieve this the mimic agent was written, which subscribes to the temperature topics from the lab, then

publishes messages once a second on the temperature mimic topic.

Bridge Agent

The Southampton MicroBroker operates within the University of Southampton network domain, and access to it is restricted by the University's data access policy (enforced by the campus firewall). The mobile phone client, is effectively any client on the Internet, using a commercial ISP (in this case a mobile GPRS provider) and as such is outside the campus administrative domain. Hence for the mobile phone client to receive data from the lab it needs to be published by a publicly visible broker. The simplest solution to this would be to make the Southampton broker publicly visible (by liaising with the campus firewall team) However this would make all of our data streams publicly visible, and could potentially allow other users on the Internet to inject data into the streams.

The solution to these problems is to bridge the data from our MicroBroker onto a publicly visible broker, in this case an Enterprise Broker run by IBM. The bridge agent, which is a component of the IBM MicroBroker, allows us to only send that data which we want to be publicly visible, and is a one-way connection, so external clients can't inject data into our private data streams.

Backup Agent

Data on the broker is inherently transient in nature (or at best retaining the last message). One aim of the lab monitoring work is to be able to review old data, so that we can consider the lab conditions when analysing data (and possibly justifying poor data), therefore we need a way of retrieving old data. The backup agent performs the "store" function of this recall, by subscribing to all the lab data streams, and writing the data from the messages into a SQL database.

Dashboard Agent

The phone client requires messages suitable for client display, and is less concerned with the raw message format. Generating these messages requires a simple transform, mapping messages from the lab topics onto dashboard topics, and also extracting the data from the lab topics, and writing it out as dashboard display texts.

Inference Agent

By combining the existing data streams, and performing some action on the data, new “inferred” streams can be generated. At this time a simple example of this is the “lab occupied” topic. To generate this, an agent listens for messages from the three PIR sensors in the lab, performs an OR operation on the result, and then adds a 30 second release delay. This provides a good approximation of lab occupation, as the raw PIR data is noisy (frequently toggling as people move round the lab), and it is rare to spend more than 30 seconds in the lab without triggering one of the PIRs.

4. Discussion

1. Effect on the chemist

The current implementation was designed as an exemplar of the technology, and to be implemented quickly, with the possibility of providing the chemist with some added value. The data provided by the current sensors is not of much direct use in real-time. However as a recallable data-set the temperature, and room access information is valuable for corroborating poor quality experimental data.

One exception of note, is that whilst at a recent meeting in Paris, discussing associated work, it was noticed that the temperature in the lab was somewhat higher than usual. This was reported back to the people working in the lab by email, who then discovered that the Air-Conditioning wasn't performing efficiently, and hence an engineer was called to rectify the problem.

This exemplar has provided a number of ideas for data that will be useful in real-time, these are currently being implemented, and some are mentioned in the future work section.

At present we don't transmit the actual experiment data, this decision was reached for two reasons. Firstly the data produced is reasonably large (and the aim is to keep MQTT messages small to minimise transmission costs), and more importantly, the display capabilities of smaller clients (such as the phone) makes a graphical display of the data of little use. More useful will be to send a message stating that the experiment has finished, and giving a Unique Identifier that can be entered into a web page to display the data, typically on a device such as a PC which would have much better display capabilities.

2. Security Concerns

Security is always a concern, especially where sensitive or important data is involved, when the

data goes across the public Internet, and where control messages are being sent to modify experimental parameters or turn devices on or off.

MQTT deliberately has a very minimalist approach to security, enabling appropriate security to be layered on top of it as required for any given application. Encrypting the message payload is an obvious first step in securing the data, which can be done using PKI certificates if required, to additionally provide signing for authentication and non-repudiation. Challenge/response security can be incorporated at the application level, by sending the challenge/response flows as MQTT messages over pub/sub. As MQTT is a protocol on top of TCP/IP, standard VPN (Virtual Private Network) products can be used to secure the connection, and hence the data flowing inside it. MQTT is often used with SSH (Secure SHell) as the VPN, but can also be used with more sophisticated VPN products such as IBM's WebSphere Everyplace Connection Manager (WECM). An element of security that is often not considered, is that of physical security. Remote monitoring and control technology such as MQTT and the message broker can be used to implement "lights-out" operations of labs, factories, oil wells, etc, with no need for anyone to be physically present at the location being monitored or controlled. Consequently, security devices such as PIR sensors and door/pressure switches can be used to raise an alert (over MQTT) if anyone unexpectedly enters a room or building. Identification technology such as RFID tagging can also be used to verify the identity of personnel who do enter or leave a controlled area, again using MQTT to alert the appropriate parties.

5. Future Work

1. Additional Data Streams.

The current data streams were chosen for their speed of implementation. Now that the concept has been proved, more chemically useful data streams need to be implemented. This will require the addition of extra sensors, for example, indication of laser running, and laser emitting would be helpful, as would publishing of experiment status from the data collection software (eg experiment start, experiment end, experiment needs human intervention). From these (and the existing data streams) inferred data streams can be added. These take data from the existing streams, perform some operation on them, and republish as a new

stream. Possible examples of this could be “un-authorised entry”, triggered by someone entering the lab, and tripping the interlock, whilst the laser is running (and hence interrupting the experiment), “Authorised user entry”, triggered by a someone entering the lab during an experiment, who knows how to override the interlock and stop it tripping (i.e. another laser scientist). If we could identify the scientist running the experiment, then this message would not be sent when they enter the lab (they'd know that they were there after all!), hence this would become an “Authorised non-operator entry”.

2. Other Laboratories

We've shown the potential usefulness of these techniques within the Surface Laser laboratory, a logical follow-on is to import these techniques into other labs. The SmartTea project is looking at ways to automate, and digitise the collection of laboratory data in the synthetic organic chemistry laboratory, with the aim to providing an end-to-end data curation solution. Part of this will be to record the environmental conditions that experiments are performed under. There is also scope to monitor reactions remotely, so that the chemist performing them can be getting on with other work. Discussions are currently under way to discover how the technology from our laser lab can be applied to their work.