A model for controlling data flow in distributed healthcare environments.

Jatinder Singh, Luis Vargas and Jean Bacon
Computer Laboratory, University of Cambridge
Firstname.Lastname@cl.cam.ac.uk

***Abstract*—Parties providing health services must have access to relevant patient data. Data sharing is common, as healthcare environments involve the collaboration of care providers, often across organisational structures. Whilst data availability is crucial to the care process, the confidential nature of medical information means that data must also be protected. To assist in the new data management concerns of distributed health systems, this paper introduces a model, built upon publish-subscribe mechanisms, for controlling the flow of data in a multi-domain homecare environment.**

I. Introduction

Information sharing is central to the provision of care, where the actions and observations of one healthcare provider are often directly relevant to another. The sensitive nature of medical information means that data must be protected. Appropriate protection is circumstantial, depending on factors including the medical condition, environmental context (e.g. emergency), the carers, patient preference, technologies (e.g. sensors), etc. The push towards a distributed care environment implies that more, possibly autonomous, entities will interact outside of a boundary of central control. As pervasive health is a highly data-driven setting, mechanisms are required to 1) share data through relevant notifications of events as they occur, and 2) actively control/protect information as it flows through the environment.

This paper describes a model to address these requirements. Much literature focuses either on access control mechanisms for data records, or on network issues such as routing and heterogeneity. Our model accounts for both, by integrating data control mechanisms into the event dissemination infrastructure itself, to *actively* control information as it flows from the source. This involves defining the conditions for transmission, and adapting messages to suit circumstance. We apply the model to a scenario for a post-operative mastectomy patient, to demonstrate its ability to encode and enforce healthcare informational constraints in response to occurrences in an active homecare environment.

II. Home healthcare specifics

Home health involves providing care services in a patient's home. Such environments increase freedom and mobility for patients, whilst reducing strain on resources (such as hospital beds). Home healthcare environments are interesting for research, as they constitute small, dynamic domains, created on demand to cater for an aspect of a patient's well-being [1]. Those requiring access to data from home environments may be mobile and transient, acting outside of larger, more static environments (e.g. a hospital). Sensor-rich homes require autonomous management of data streams, considering context as part of the information aggregation and dissemination process. As each home domain is unique, it requires a custom management policy based on circumstances such as patient details, personal preference, geographic location, available technologies, available finances, insurance, etc.

A. Confidentiality of medical data

Privacy concerns data, referring to the circumstances for which particular information may be disclosed [2]. Privacy in healthcare is expected, and imposed by oath, codes of conduct, and law. Improvements in communication and data storage (aggregation) technologies tend to increase the risk of incorrect data disclosure by providing more parties the potential to access greater amounts of information [3]. The problem is exacerbated in the realm of pervasive health, as it is a highly collaborative setting, involving interactions between users (human) and other entities (e.g. sensors), in an environment *without* central control.

B. Domains and Entities

Our work is built upon the concepts of domains [4] and entities. We define a *domain* as a named grouping structure with a particular function/motivation. A domain might represent an organisation, such as doctor's surgery, or other groupings, such as an association with a patient's home. Each domain maintains its own policy for controlling interactions with entities and other domains. *Entities* are the actors in the system, introducing new data through actions and events. Examples of entities include doctors, carers, patients and sensors, that interact with the environment through applications/interfaces. Entities may be grounded (registered) in a domain, meaning the domain holds some of their credentials, to avail or validate upon request [4]. Depending upon policy, entities may perform actions in other domains for which they are not registered.

Data availability/sharing is a requirement for the proper provision of healthcare services; however, data must also be properly protected. Therefore, mechanisms are required to allow domains in pervasive health environments to control the circumstances in which data is released.

III. Middleware

Our focus is on middleware to control data flow in a healthcare environment. Middleware is a software layer that

lies between a physical (computing and network) infrastructure and various applications. In pervasive health environments, care services involve interactions both within and across domains. Middleware must account for this loosely-coupled environment, where the data models of entities and domains differ. As middleware acts as a level of indirection between applications, it provides a central point for policy enforcement.

A. Event-based middleware

Modern healthcare environments are, by nature, *event-driven*. That is, parties generate events (directly or via actions), for which others receive notification. Events themselves are data-rich, containing attributes with associated values [5].

To control data flow in a healthcare environment, we focus on an event-based middleware built around the *publish/subscribe* paradigm [6]. In this paradigm an entity, through an application/interface, takes the role of an event producer and/or an event subscriber. Subscribers subscribe to event types of interest (e.g. a prescription) and may optionally specify a condition (filter) on the event's content (e.g. drug='penicillin'). An event producer publishes events independently of a subscriber. If a published event matches a subscription (type and condition), the middleware delivers the event to the subscriber. An *event broker* (broker) manages the middleware functionality, by routing published events to subscribers, i.e. other brokers or entities.

B. Event-based database middleware

Databases constitute an obvious point for information dissemination management. We have extended a database system to provide event-based middleware support [7], to achieve information sharing systems that are simpler to deploy and maintain. Grouping security, configuration, and recovery tasks for database and messaging operations under the same interface provides a single point of control – the database system. Further, event-based middleware functionality benefits from the native reliability and recoverability features of database systems, storing event related information through the same means as general data.

An integrated event-based middleware and database system may function as an event broker. In our current prototype, a connection is made to a broker to utilise middleware functionality. The form of a publication is in reference to a named event type and schema (set of attribute name/datatype pairs), thus allowing event types to be advertised to other applications in the system. The middleware supports the reliable delivery of events through the use of acknowledgement receipts. Subscribers may specify an optional condition (filter) defining the circumstances in which they receive an event. An advantage of using a database system in a data control model is the expressiveness of in-built languages (e.g. SQL) that support both in-built and user-defined functions. This allows powerful and fine-grained conditional clauses that may reference event content (data), context and external services, which are used to define the constraints in our data control model.

IV. Data Flow Control Model

In general, event-based middleware focuses on delivering an event to all interested entities, i.e. all event subscribers whose subscriptions match an event. However, data in the healthcare environment must be protected. The premise of our work is that data should be controlled as it flows from the source, preferably transferred on a need-to-know basis. This section details a model allowing domains to define, in policy, the conditions under which data is released.

A. Broker network

Each domain uses a broker to maintain its policy for controlling the flow of data within and between domains. Thus, a broker maintains a set of 1) known event types, 2) active subscriptions and 3) information control policies.

As depicted in Figure 1, domains are interconnected via their event brokers, where a broker in one domain subscribes to events occurring/received in another. This is for two reasons. Firstly, to update its local data store, providing access to relevant information that may be subsequently queried. Secondly, to actively forward the information to other subscribed parties. Communication in this model occurs directly between the broker and the subscriber, meaning the type and content of events transmitted is subject to the policy of the broker.

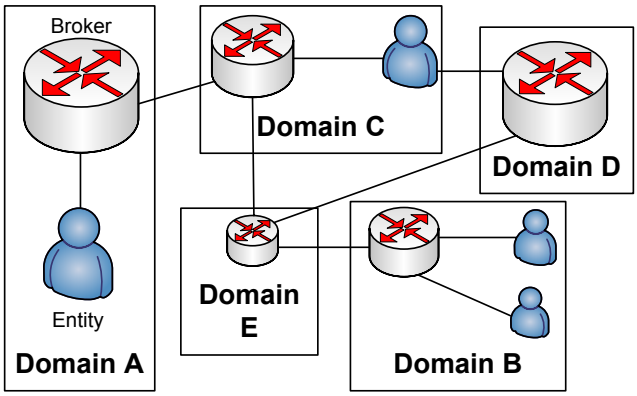

Fig. 1. An example broker network showing communication across domains.

B. Data flow control

The following provides the basis for controlling data flow:
Transformation. A broker has the ability to transform an event, which can involve functions that modify, mask or summarise various event attributes, and/or perform a type conversion. Type conversion can include information not present in an original event (e.g. more attributes, or other related information), or degrade an original event (e.g. hiding sensitive attributes and type names). Schemas are provided to subscribers so that they are aware of the types of the events they will receive. Common transformations may be defined as 'mapping functions', allowing reuse by other policy elements.
Conditional Control. Conditional clauses are central to our model, as they define the circumstances in which the control mechanisms apply. The data handling and function capabilities provided by our database middleware allow such clauses

to reference event content, user credentials, environmental context, stored data and results of external functions.

Using the above, our model uses policy definitions based on the following to control the flow of data:

a) Publication of new events. Upon receipt of a certain event, a broker may publish one or more 'new' events. These may be a transformed version of the original, an event with some derived information, or an independent event to notify of an occurrence. Policy defines the method for creating new events and the conditions for doing so. The created events are subject to the same subscription matching, data transformation and dissemination processes as general publications.

b) Subscriber Control. This allows policy to target subscribers holding particular credentials. This customisation is realised through the following two operations:

Imposed conditions. Our model, like many publish/subscribe systems, allows a subscriber to specify the events they receive through a filter on event content. However, we allow policy to impose additional constraints to specify the circumstances where a subscriber receives an event. To ensure adherence, these conditions override any subscriber specified preference. This is done silently to avoid leakage of any sensitive information encoded in the policy itself.

Custom data transformation. This refers to the ability to customise (transform) an event to suit a subscriber with particular credentials in defined circumstances. The transformation process is similar to that of an event publication, except that a transformed event is sent directly to the subscriber to which the policy applies, rather than being published as 'new'.

Policy is central to our model as it drives the control mechanisms. Our model specifies policy in XML, as it is a readily understood standard. The next section details application of this model to a healthcare scenario.

V. Scenario – Post-Operative Care for Mastectomy

To test the expressiveness of our model in managing health information, we consulted nursing manuals, medical codes of practice and legislation, and adapted the requirements to a home healthcare environment. This section focuses on aspects of care for a post-operative mastectomy patient – a common treatment for breast cancer. Such patients may experience a number of conditions, such as pain, swelling around the wound, seroma, stiffness, soreness and chording [8]. Caring for a patient in this scenario may involve prescriptions for antibiotics or pain relief, seroma drainage and physiotherapy [8].

In this paper, we focus upon data management aspects regarding the administration of prescriptions. A key aspect of care for a post-operative mastectomy patient is pain management. Generally, a nurse is authorised to prescribe a licenced medicine for any condition within their competence [9]. In specific cases, such as those of acute post-operative pain, a nurse may prescribe certain controlled drugs, such as morphine derivatives; useful in home scenarios where a doctor is not present. Prescriptions for controlled drugs have strict requirements, both in form and handling [10]. Further, specific bodies are responsible for monitoring the supply of such drugs, though legislation focuses upon the drug supplier, where monitoring should maintain patient privacy [11].

A. Data Flows and Policy

We identify four domains that interact as part of the prescription process for post-operative mastectomy: 1) the Surgery directly responsible for the care of the patient, 2) the Pharmacy that the surgery uses to supply drugs, 3) the patient's home environment, and 4) the Auditor, responsible for monitoring the supply of controlled drugs[1].

We consider two entities in this scenario, the homecare nurse and the patient's physician, both grounded (holding credentials) in the Surgery domain. The nurse, whilst caring for the patient at home, may decide to prescribe a medicine, perhaps a controlled drug if the situation warrants. This involves the nurse publishing (e.g. though a PDA connected to the Home domain) a `prescribe` event, which includes information of the drug, dosage, symptoms, notes and observations. The physician requires notification when controlled drugs are supplied, as this might indicate a condition of concern.

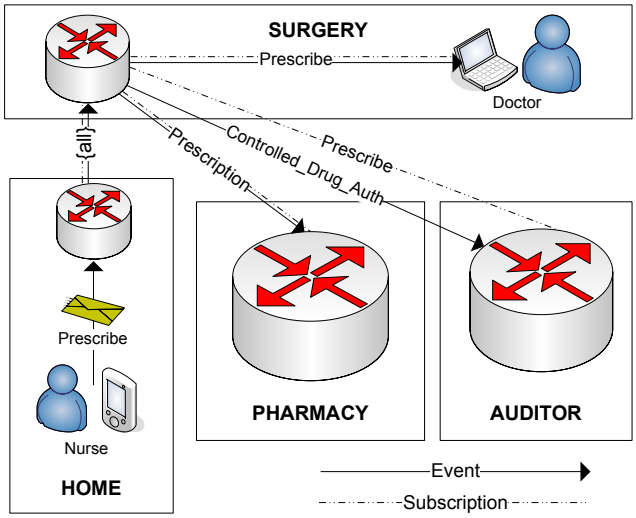

Fig. 2. Subscription based event flow for a prescribe event.

Figure 2 depicts the subscriptions (interests) of the parties and the event types received from the publication of a `prescribe` event. Here, most subscribers receive an event type matching their subscription. The exception is the Auditor, who receives a modified event with the patient details removed. The Surgery is the body legally responsible for the patient, and thus subscribes to all events occurring in the Home domain – to be aware of all occurrences within that environment. We model this scenario so that the Surgery releases data to other domains, because 1) it is responsible for the patient, 2) often information will need to be processed/verified before transmission elsewhere, and 3) this extra level of indirection makes the patient less identifiable. Whether a Surgery acts as a proxy for a home domain is a question of design/circumstance – our model can support either representation.

[1]The Auditor is modelled as a domain as it might be a group (e.g. a governmental structure) rather than an individual entity.

```
<subscriber_control>
  <event_type>prescribe</event_type>
  <credential>physician</credential>
  <condition>
     managesPatient(user, patient)</condition>
</subscriber_control>
```

Fig. 3. Policy imposing a condition to ensure that physicians receive only that information regarding patients that they manage.

```
<subscriber_control>
  <event_type>prescribe</event_type>
  <credential>auditor</credential>
  <condition>isControlledDrug(drug)</condition>
    <publish publish_type="controlled_drug_auth">
      <condition></condition>
      <field name="drug_name">drug</field>
      <field name="dosage">dosage</field>
      <field name="prescribedby">
         getCarerCredentials(carer)</field>
      ....
    </publish>
</subscriber_control>
```

Fig. 4. Policy for informing the Auditor that a controlled drug was authorised. This involves imposing a condition and performing a transformation that removes patient specific information.

Data flow policy. Here we detail the data policy for the Surgery domain. In this example the physician wishes to be notified when controlled drugs are prescribed. Surgery policy allows a physician to subscribe to any `prescribe` event concerning a patient they manage. Figure 3 illustrates policy to enforce this condition. Upon subscription, the physician is free to specify, for example, that they only require notifications for controlled drugs; however, the policy ensures, through use of a function, that a proper management relationship exists.
Legislation requires that the supply of controlled drugs must be monitored, without compromising patient privacy. In this example, the Auditor domain subscribes to `prescribe` events, as it has an interest in drug supply. However, the policy, as per Figure 4, involves transforming such an event into a `controlled_drug_auth` event, which serves to remove patient identifiable information. This transformation is not republished, but transmitted directly to the Auditor, as it is the only party that should receive information in this form.
A `prescribe` event does not itself constitute a legal prescription, as a prescription requires patient specifics (e.g. current address), and should not contain information of symptoms or carer's notes. As such, the Pharmacy subscribes to `prescription` events, which encapsulate all information constituting a proper prescription. Figure 5 shows domain policy for republishing a `prescribe` event into a `prescription` event, which utilises user-defined mapping functions to perform the transformations (omitted for want of space). As controlled drugs have special prescription requirements, the transformation function applied depends upon the particular class of drug prescribed.

```
<message_control>
  <event_type>prescribe</event_type>
  <publish publish_type="prescription">
    <condition>isControlledDrug(drug)</condition>
    <mapping>to_controlled_prescription</mapping>
  </publish>
  <publish publish_type="prescription">
    <condition>isGeneralDrug(drug)</condition>
    <mapping>to_prescription</mapping>
  </publish>
</message_control>
```

Fig. 5. Transformation policy for republishing a Prescribe event into a proper Prescription, as appropriate for the class of drug.

VI. Related work and conclusion

We expect this work complements other control mechanisms, such as attribute encryption models [12], and access control schemes, features of which would allow more detailed control polices. Our model is related to workflow, though our intention is *not* to prescribe action sequences for entities. Rather, we provide mechanisms to actively control, according to circumstance, information as it flows from the source – through means of adaptation and/or conditional access. Our next steps include creating usable auditing mechanisms for tracking information flows, and integrating sensor (streaming) technologies into the environment, to autonomously manage event patterns from multiple streams. The model presented is able to encode and enforce real healthcare informational constraints, as derived from nursing manuals and legislation; thus, we feel it an appropriate base for further investigation into the data control issues of pervasive healthcare environments.

Acknowledgments

EPSRC GR/C53719 CareGrid and Microsoft Research supports Jatinder Singh. Luis Vargas is supported by CONACYT.

References

[1] J. Singh, J. Bacon, and K. Moody, "Dynamic trust domains for secure, private, technology-assisted living," in *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*. IEEE Computer Society, 2007, pp. 27–34.
[2] World Medical Association, "WMA declaration on ethical considerations regarding health databases," http://www.wma.net/e/policy/d1.htm.
[3] R. J. Anderson, "A security policy model for clinical information systems." in *IEEE Symposium on Security and Privacy*, 1996, pp. 30–43.
[4] J. Bacon, "Expectations and Reality in Distributed Systems," in *Proceedings of IASTED International Conference on Parallel and Distributed Computing and Networks*, Insbruck Austria, Feb. 2005, pp. vii – xiv.
[5] G. Muhl, L. Fiege, and P. Pietzuch, *Distributed Event-Based Systems*. Springer, 2006.
[6] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, 2003.
[7] L. Vargas, J. Bacon, and K. Moody, "Integrating Databases with Publish/Subscribe," in *Proceedings of the 4th Int'l Workshop in Distributed Event-Based Systems (DEBS'05)*. IEEE Press, June 2005, pp. 392–397.
[8] G. J. McNeal, *AACN Guide to Acute Care Procedures in the Home*. Philadelphia, PA, 19106: Lippincott Williams & Wilkins, 2000.
[9] Department of Health (UK), "Safer management of Controlled Drugs," http://www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_074513, 2007.
[10] Royal Pharmaceutical Society of Great Britian, "Medicines, Ethics and Practice: A guide for pharmacists and pharmacy technicians)," http://www.rpsgb.org.uk/pdfs/MEP31s1-2b.pdf, July 2007.
[11] "The Controlled Drugs (Supervision of Management and Use) Regulations 2006 (UK)."
[12] J. Bacon, D. Eyers, K. Moody, and L. Pesonen, "Securing publish/subscribe for multi-domain systems," in *Middleware '05*, Nov. 2005, pp. 1–20.