Services, processes and policies for digital health: FHIR® case study

Zoran Milosevic
Deontik Pty Ltd
Australia
zoran@deontik.com

Andy Bond
NEHTA,
Australia
Andy.Bond@nehta.gov.au

Abstract — **This paper provides a number of recommendations for expressing the behavioral semantics of a digital health enterprise. This includes the specification of policies, processes and services, with a particular focus on the RM-ODP enterprise and computational concerns. We believe that these behavioral aspects are needed to complement well developed expression of information semantics. The paper is motivated by the need for better approaches to expressing behaviour in the emerging FHIR® standard and providing a solid architecture framework to practitioners when using FHIR. However, the concepts and ideas are more general and can be applied to other technologies when building digital health solutions.**

Keywords - e-health; digital health; RM-ODP; FHIR

I. INTRODUCTION

Many e-health initiatives over the last decade or so have been concerned with modelling information about patients and their healthcare problems, conditions or treatment. This information has been captured, recorded and used as part of electronic health records or other applications. Such applications have been developed with a focus on supporting information semantics, a key aspect needed to replace paper-based systems, thus enabling many downstream applications to become part of a technology enabled healthcare infrastructure.

In order to support interoperability between various vendor offerings, several healthcare informatics standards were developed, mostly from the Health Level Seven (HL7) standards body [1]. The early standard was V2 messaging, providing protocols and data elements for applications to use when exchanging data, e.g. pathology orders and response messages from laboratory information systems. Other HL7 standards capture healthcare information in clinical documents, in particular HL7 Clinical Document Architecture (CDA). CDA was based on the HL7 Reference Information Model (RIM) standard using UML as the underlying modelling framework. Other informatics standards are openEHR [2] and CIMI [3] based on different reference models. While CDA and openEHR are mostly used for 'data at rest' applications, HL7 v2 messaging has wider use in data exchange applications or clinical transactions.

Although these standards have been supported by many health IT vendors, the development of e-health applications has never been a trivial task. This is because of the specific modelling framework developed by HL7, which was quite complex, requiring many extensions and adaptations to reflect specific needs of different healthcare environments. The steep learning curve for developing HL7 models and implementations using these standards meant long implementation and deployment times, yielding significant costs.

The most recent HL7 standard, Fast Health Interoperability Resources (FHIR), provides a new approach to developing e-health applications, such as medication management, electronic referrals, clinical decision support systems and medical records. FHIR is based on the commonly used web-based technologies, allowing ease of implementation, while leveraging experience from many years of different e-health implementations, providing a rigorous mechanism for exchanging data between healthcare applications.

The main modelling concept in FHIR is called a *resource*, which is an information component that can be used to exchange and/or store data to support building e-health solutions. Resources cover clinical and administrative components and also include some of the infrastructure components. Many e-health projects are now looking at using FHIR for new e-health applications and the standard is gaining significant interest and adoption. The mainstream resources are focussed on clinical content models but it is increasingly recognised that many e-health applications require explicit support for defining and implementing services and processes (workflows), with additional support for the expression of enterprise policies that apply to those involved in delivery of healthcare, including patients themselves. The existing focus on information semantics thus needs to be extended to better support behavioural semantics, as suggested in several e-health interoperability frameworks [5][6][7].

This paper provides suggestions how the RM-ODP concepts can be used to provide behavioural support for FHIR. The paper also identifies a number of new concepts published in the recent RM-ODP Enterprise Language (EL) standard, revised in 2015 [9], and shows how these concepts can provide further semantics support for describing the complex digital health environment, compared to that which was identified in our earlier service semantics [8]. In order to provide a self-contained description of ODP-based behavioural semantics, the paper includes a selected set of related concepts defined in the ODP foundations standard, as revised in 2010 [10]. These concepts will be discussed in the context of the FHIR specification, with the aim to provide several architectural proposals for more expressive behavioural semantics.

The next section provides a brief description of the FHIR standard. Section III introduces a set of foundational behavioural concepts, based on the foundational concepts in RM-ODP. Section IV describes how these concepts can be refined for the needs of enterprise modelling, required to precisely specify the environment within which an e-health system is to be deployed. Section V discusses behavioural semantics associated with the life-cycle of information components. Section VI describes key concepts for modelling computational interactions in an e-health system. Section VII outlines engineering and technology concepts. Note that the last four sections also discuss existing FHIR resources and identify new components (or resources) that may be needed for better support of services, processes and policies. Section VIII provides discussion.

II. FAST HEALTHCARE INTEROPERABILITY RESOURCES

This section introduces key concepts from the May 2016 FHIR release [4]. Note that FHIR is not yet a standard and is still in a state of development. A number of drafts have been made available but these can see significant changes between releases. Organisations have already released production systems based on these drafts but clearly must understand any implications when feature sets may change as the standard emerges.

A. *FHIR Resource*

The FHIR Resource is the basic building block used in information storage and exchange within and between digital health applications. A resource consists of predefined data elements of specific data types, some of which point to other resources. It uses a structured, hierarchical representation of its elements, supporting human readability, and a corresponding alternative logical specification in UML, as well as an implementable representation in XML and JSON (Figure 1).

Resources have a wide range of uses, from pure clinical content such as care plans and diagnostic reports through to pure infrastructure such as message headers and conformance statements. In clinical content form, they are granular enough to allow the composition of clinical content meeting a variety of use cases with the ability to extend to incorporate features either not envisioned or otherwise deemed extraordinary to most use.

B. *Resource Types*

The resources are classified into 6 sections (Figure 2):

Clinical, which provide core clinical information foundations - focused on the content of the provider/patient encounter. They are classified in terms of General, Care Provision, Medication & Immunisation and Diagnostics. Examples are *DiagnosticReport* and *Observations* that support Clinical Findings, such as Laboratory Results; *Condition* resource to support a patient's Past Medical History, e.g. Diabetes, Congestive heart failure, etc.

Identification, which provide entity foundations for the care process including individuals (such as Patient and Practitioner), Groups (such as Organisation and Group), Entities and Devices.

Workflow, for managing the healthcare process, covering subcategories of patient management, workflow specific resources and decision support.

Financial, for billing and payment processes.

Conformance, for managing specification, development and testing of FHIR solutions.

Infrastructure, providing general functionality and resources for internal FHIR requirements.

C. *Observations: current FHIR specifications*

The current FHIR emphasis is on representing information components related to the delivery of healthcare. There is no support for explicit service modelling, apart from the simple REST style of interaction, which is mostly concerned with the lifecycle and actions associated with resources. This is not to say that services are not part of FHIR, rather their manifestation is through the packaging of behaviour intent within resources and relying upon the obligations of service execution to be carried by the receiving FHIR servers. More recently, workflow related resources have begun to emerge within FHIR, at this stage allowing for the specification of tasks. The broader aspects of workflow coordination and even this task start are not well developed yet, as shown with a recent FHIR maturity ranking of 0 (documented next to the resource within the FHIR specification). They lack the workflow style of behaviour, such as data flows and control flows, rather they focus on the representation of workflow artefacts, such as task, as information components.

The FHIR Implementation Guide (IG) is a FHIR mechanism to provide instructions on how to make use of FHIR in a particular problem space. These guides, include a collection of FHIR profiles defining a selection of localised FHIR resources, additional elements as needed, a description of how resources and APIs map to local requirements, etc. These guides, once created, are intended for developers and there are currently several such IGs. An Implementation Guide can be as general or detailed as needed. Like all of FHIR, an IG is ultimately just another resource with mandatory and optional content. The foundation IG resource could in turn be profiled and modified for local use. An IG does not imply any particular architecture paradigm but may include profiles of resources that orient the problem space to messaging (MessageHeader and OperationOutcome), documents (Composition and DocumentManifest), or business services (HealthcareService)

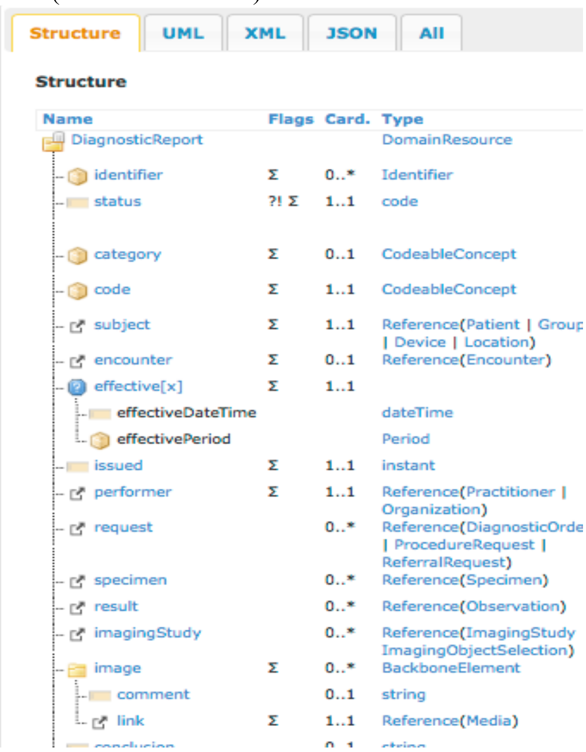

Figure 1: Diagnostic report resource (fragment)

Clinical

General:

- AllergyIntolerance 1
- Condition (Problem) 2
- Procedure 1
- FamilyMemberHistory 1
- ClinicalImpression 0
- DetectedIssue 1

Care Provision:

- CarePlan 1
- CareTeam 0
- Goal 1
- Protocol 0
- ReferralRequest 1
- ProcedureRequest 1
- NutritionOrder 1
- RiskAssessment 0
- VisionPrescription 0

Medication & Immunization:

- Medication 1
- MedicationOrder 1
- MedicationAdministration 1
- MedicationDispense 1
- MedicationStatement 1
- Immunization 1
- ImmunizationRecommendation 1

Diagnostics:

- Observation 3
- DiagnosticReport 3
- DiagnosticOrder 1
- Specimen 1
- Sequence 0
- BodySite 0
- ImagingStudy 2
- ImagingExcerpt 0
- ImagingObjectSelection 1

Identification

Individuals:

- Patient 3
- Practitioner 1
- PractitionerRole 0
- RelatedPerson 1

Groups:

- Organization 1
- HealthcareService 1
- Group 1

Entities:

- Location 1
- Substance 1
- Person 1
- Contract 0

Devices:

- Device 1
- DeviceComponent 1
- DeviceMetric 1

Workflow

Patient Management:

- Encounter 1
- EpisodeOfCare 1
- Communication 1
- Flag 1
- Appointment 1
- AppointmentResponse 1
- Schedule 1
- Slot 1

Workflow:

- Task 0
- Order 0
- OrderResponse 0
- CommunicationRequest 1
- DeviceUseRequest 0
- DeviceUseStatement 0
- ProcessRequest 0
- ProcessResponse 0
- SupplyRequest 0
- SupplyDelivery 0

Decision Support/Quality Measurement:

- DecisionSupportServiceModule 0
- DecisionSupportRule 0
- GuidanceResponse 0
- Library 0
- OrderSet 0
- Measure 0
- MeasureReport 0
- ModuleDefinition 0

Infrastructure

Information Tracking:

- Questionnaire 0
- QuestionnaireResponse 2
- Provenance 1
- AuditEvent 2

Documents & Lists:

- Composition 2
- DocumentManifest 1
- DocumentReference 2
- List 1

Structure:

- Media 1
- Binary 1
- Bundle 2
- Basic 1
- Linkage 0

Exchange:

- MessageHeader 2
- OperationOutcome 2
- Parameters 1
- Subscription 1

Conformance

Terminology:

- CodeSystem 0
- ValueSet 3
- ConceptMap 2
- ExpansionProfile 0
- NamingSystem 1

Content:

- StructureDefinition 2
- StructureMap 0
- DataElement 1

Operations Control:

- Conformance 2
- OperationDefinition 1
- SearchParameter 1
- CompartmentDefinition 0

Misc:

- ImplementationGuide 0
- TestScript 0

Financial

Support:

- Account 0
- Coverage 0
- EligibilityRequest 0
- EligibilityResponse 0
- EnrollmentRequest 0
- EnrollmentResponse 0

Billing:

- Claim 0
- ClaimResponse 0

Payment:

- PaymentNotice 0
- PaymentReconciliation 0

Other:

- ExplanationOfBenefit 0

Figure 2: FHIR Resources (May 2016 Release)

III. Foundational Concepts for Behaviour

This section introduces a selected set of foundational concepts for behaviour of relevance for FHIR. They are presented as they are defined in the RM-ODP foundations standard [10], structured in several categories. It is important to understand that some of these concepts can be used as defined below, or they can be refined to add further semantics associated with a particular viewpoint. Thus, they need to be considered from different stakeholders concerns, which can be supported by applying different ODP viewpoints [11]. Some categories support the specification of the healthcare environment in which a health IT system is to be built, namely organisational structure and policies, while others are of relevance for describing health IT information and applications.

A. Basic modelling concepts

Object is one of the basic modelling concepts, representing a model of an entity from the real-world, which can be physical (e.g. a mobile phone), a human (e.g. Bob, the patient) or an organisation (e.g. a hospital) or a more abstract entity (e.g. discharge summary). An object is characterised by behaviour and dually by its state. When considering one system from different ODP viewpoints, one can distinguish between enterprise objects identifying entities in business (see section IV), information objects providing information about enterprise objects for example (see section V), or computational objects, which are building blocks for applications (see section VI). FHIR is concerned with information objects at present. The enterprise concerns are either treated implicitly, or explicitly captured in natural language descriptions in IGs.

Action is defined as anything that may happen. Every action of interest for modelling purposes is associated with at least one object [10]. An action that involves participation from the object's environment is referred to as an *interaction*, e.g. sending a message to another object. Again, actions can be enterprise, information or computation, depending on the viewpoint. Each of the objects involved in the interaction plays a particular *action role* characterised by the information it contributes or accepts and by whether or not it originated the action [11]. An action that does not involve participation of the object's environment is referred to as an *internal action*, e.g. the model of sudden failure of a computer.

Behaviour is defined as a collection of actions with a set of constraints on when they may occur. Examples of behaviours are sequentiality, concurrence or real-time constraints, which can be used to describe business processes in the enterprise viewpoint or RPC interactions from the computational viewpoint.

Event is defined as the fact that an action has taken place. When an event occurs, the information about the action that has taken place becomes part of the system and may be thus subsequently communicated in interactions. Such communication is called an event notification. It carries information about the event from the object that performs or observes it, to other objects that have a need to take action as a result of the event [10]. In the computational viewpoint, the concept of event is used frequently in the context of a publish-subscribe pattern.

Interface is an abstraction of the behaviour of an object that consists of a subset of the interactions of that object. Note that an interface forms part of the objects' behaviour considering only the interactions in that interface, while hiding all other interactions. In the computational viewpoint, an interface is often used as a way for a computational object to offer its services to the environment, as in an SOA style architecture.

The *state* of an object is the condition of an object at a given instant in time that determines the set of all sequences of actions (or traces) in which the object can participate. Note that state changes are effected by actions.

B. Specification concepts

Object *composition* is a combination of two or more objects yielding a new object, at a different level of abstraction. The characteristics of this new object, called a composite object, are determined by the objects being combined and by the way they are combined. The behaviour of the composite object is the corresponding composition of behaviour of the component objects. Examples of composition techniques are sequential composition, interleaving, choice and hiding of actions.

Role is a formal placeholder in the specification of a composite object, identifying those behavioural aspects of some component object serving as constraints on an actual object in an instance of the composite. Therefore, in order to satisfy the specification, the actual object is required to exhibit the specified behaviour and thus it fulfils the role in the instance of the composite. This is a generic definition of role, with specific types being action role in interaction, role in a (computational) binding (see section VI), or a community role (see section IV.A).

C. Service concepts

Service is defined as a behaviour, triggered by an interaction between provider and consumer objects, that adds value for its users by creating, modifying or consuming information; the changes become visible in the service provider's environment. Note that the provider's environment includes the service user and thus that provision of a service involves some kind of commitment by the provider to stand by its actions. As indicated in [8], a frequently used term 'service offer' is a necessary but not sufficient condition for a service to be instantiated. Participation of a service user in the collaboration is also required,

A service can be composed from other services and it can also have relationships with other services. A special kind of relationship is an 'is a' relationship typically used to construct relationships between elements in a taxonomy, e.g. an orthodontist service is a dental service.

Note that this definition refers to the concept of service as an instance of a real-world thing that involves interactions between service users and service providers.

It implies that there is an implicit or explicit agreement between them (i.e. a service contract). There may also be a service description as a way of expressing a service provider offering to potential service users, i.e. what value the service will deliver. In RM-ODP, a service description can be expressed using the concept of template, defined as 'the specification of the common features of a collection of <X>s in sufficient detail that an <X> can be instantiated using it', e.g. concrete classes in object oriented languages such as Java.

D. Organisational concepts

The foundational concepts also include several organisational concepts that are useful to define interaction boundaries, which is important when defining different scopes of interoperability as discussed in NEHTA Interoperability Framework [5].

<X> Group is defined as a set of objects with a particular characterizing relationship <X>. The relationship <X> characterizes either the structural relationship among objects or an expected common behaviour of the objects. An example of specialized groups is addressed group, i.e. a set of objects that are addressed in the same way.

Configuration (of objects) is defined as a collection of objects able to interact at interfaces. Note that this concept will be further refined in the enterprise language when introducing the definition of community (section IV.A).

The specification of a configuration may be static or may be defined in terms of the operation of dynamic mechanisms which change the configuration, such as *binding* and *unbinding*. Binding plays an important role in the computational viewpoint (section VI) for connecting computational objects but also in the enterprise viewpoint in the context of the process of fulfilling community roles (section IV).

<X> domain is defined as a set of objects, each of which is related by a characterizing relationship <X> to a controlling object. Note that every domain has a controlling object associated with it. The controlling object can determine the identities of the collection of objects which comprise the associated domain. The controlling object may communicate with a controlled object dynamically or it may be considered to have communicated in an earlier epoch of the controlling object. Generally, the controlling object is not a member of the associated domain.

E. Policy concepts

Policy is defined as a constraint on a system specification foreseen at design time, but whose detail is determined subsequent to the original design, and is capable of being modified from time to time in order to manage the system in changing circumstances. Policies can be applied in any viewpoint, e.g. an enterprise delegation policy, a computational persistence policy or an engineering scheduling or quality support policy. Enterprise policies may be expressed in terms of obligations, permissions or prohibitions.

Rule is defined as a constraint on a system specification. Where appropriate, a rule can be expressed as an obligation, a permission or a prohibition.

Obligation is a prescription that a particular behaviour is required. An obligation is fulfilled by the occurrence of the prescribed behaviour.

Permission is a prescription that a particular behaviour is allowed to occur. A permission is equivalent to there being no obligation for the behaviour not to occur.

Prohibition is a prescription that a particular behaviour must not occur. A prohibition is equivalent to there being an obligation for the behaviour not to occur.

Contract is an agreement governing part of the collective behaviour of a set of objects. A contract specifies obligations, permissions and prohibitions for the objects involved.

IV. Enterprise Viewpoint

In the enterprise viewpoint, one is concerned with understating and describing an organisational and social environment, in which an IT system is to be deployed. In a healthcare context such systems are referred to as health IT systems, or digital health. The RM-ODP Enterprise Language standard [9] (ODP-EL) provides a number of modelling concepts and structuring rules to provide this description in terms of precise environment semantics in which it is to be built. In many respects this is similar to the concerns of social ontologies, such as UFO-C [8], but the focus here is on *prescriptive*, rather than *descriptive,* specifications, as with ontologies. Both approaches aim at establishing a common understanding of the environment, but RM-ODP provides a reference architecture for building systems, e.g. when developing e-health solution architectures. These should be technology neutral, supporting the description of its realisation in technologies such as FHIR, openEHR or CIMI.

An enterprise specification defines the purpose, scope, and policies of an ODP system and it provides a statement of conformance for system implementations. The purpose of the system is defined by the specified behaviour of the system while policies capture further restriction on the behaviour between the system and its environment or within the system itself related to the business decisions by the system owners. An enterprise specification also allows the specification of an ODP system that spans multiple domains and is not owned by a single party, and specification of the collective behaviour of a system that is divided into independently specified and independently working subsystems [9]. This generality places greater emphasis on the expression of correct or normal behaviour and on the chains of responsibility involved in achieving it. For example, the advent of service-oriented and cloud computing has led to the need to specify business rules and behaviour in a way that clearly describes obligations, permissions, authorizations and prohibitions, as well as the accountability of each of the objects involved in an enterprise specification. This involves the expression of the so-called deontic aspects of the behaviour of the system, and of the accountability of the objects involved.

Key ODP-EL concepts are introduced next. They include relevant foundational concepts and the refinement of these for the specific purpose of enterprise modelling. Note that this description includes a number of new modelling concepts proposed in the ODP-EL standard revision [9], structured in terms of different categories.

A. Community concepts

Community is the first-class modelling concept in the ODP-EL, used to describe the organisational or social environment. It defines how a set of participants should behave in order to achieve an objective. To make the rules reusable, they are expressed in terms of interactions between roles in the community, decoupling their definition from the details of the available resources and the responsibilities in a specific situation [11].

Formally, *community* is defined as a configuration of (enterprise) objects formed to meet an objective [9]. The objective is expressed in a contract, which expresses how this objective can be met by defining (community) roles and required interactions, assignments of (enterprise) objects to the (community) roles, and (enterprise) policies governing their collective behaviour. Note that we introduced the specific usage of the foundational concepts defined in section III as enterprise viewpoint concerns, by using brackets to qualify how these concepts can be refined in the enterprise language. ODP-EL also provides a normative definition of *objective* (of an <X>) as a practical advantage or intended effect, expressed as preferences about future states. ODP-EL emphasizes the need to express an objective in measurable terms.

In order to support more complex, cross organisational interactions, the concept of federation is introduced as a special kind of community. Formally, *<X> federation* is defined as a community of <x> domains. Note that in enterprise terms, various policies can be administered by the controlling object over the domain.

Further, ODP-EL includes the concept of *community object* to support hierarchical organisational structures, defined as a composite enterprise object (section III.B) that represents a community. Essentially, a community object is an abstraction of a community that can be used to describe complex organizational structures, both hierarchical and federation structures [11].

B. Behaviour concepts

Although community is a structuring concept, it also describes a focus for many aspects of behaviour. In general, behaviour includes both basic behaviour, in terms of processes, steps and actions, and any associated deontic or accountability mechanisms.

Typically, behaviour in a community can be described in terms of a composition of (business) processes addressing separate business concerns [11]. Therefore, it is a business process in its own right, i.e. community process.

The ODP EL defines a (business) *process* as a collection of steps taking place in a prescribed manner. Formally, *step* is an abstraction of an action (see section III), used in a process, that may leave unspecified some or all of the objects that participate in that action.

Thus, a process is an abstraction of a behaviour, and so shares any objectives defined for that behaviour. Note that the prescribed manner may be a partially ordered sequence of steps. An enterprise specification may define types of process and may define process templates. Further, a process specification can be a workflow specification.

Note that the community process is typically parameterised by community roles. This is similar to the use of BPMN partitions, but is more adequate to support a hierarchy of roles, reflecting organisational structures within a community, as will be described later on.

An *active enterprise object* is defined as an enterprise object that is able to fill an action role (section II). In other words, it is an enterprise object that can be involved in some behaviour. It is important to note that the behaviour of active enterprise objects is constrained by deontic and accountability concepts, described later in this section.

Behaviour within a community can also address how enterprise objects (filling the community roles) participate in interactions. An enterprise object can participate in performing an action, in which case it is termed an *actor* with respect to that action. An enterprise object is referred to as an artefact with respect to an action if it is not an active participant within the action. Finally, an enterprise object can be essential to an interaction, requiring allocation, or may become unavailable, in which case it is referred to as a *resource (with respect to an action).*

Note that a special kind of enterprise object is *party*, modelling a natural person or any other entity considered to have some of the rights, powers and duties of a natural person. A party can have intentions and is accountable for their actions. It is one of the accountability concepts described later in this section.

C. Deontic concepts

The latest revision of the ODP-EL introduces a number of new modelling concepts related to the expression of constraints on enterprise objects in a community. These concepts are of particular value when describing enterprise policies, some of which express the concepts of accountability, which are important for any organisation with a specific focus in healthcare. In order to preserve the completeness of this fragment of the EL specification, and highlight the value of these new concepts, we list the deontic concepts, as defined in [6].

Firstly, the concept of *deontic token* is introduced as a way to support realising permissions, prohibitions and obligations, as constraints on an object's behaviour. Formally, a deontic token is defined as a special kind of enterprise object which expresses a constraint on the ability of an active enterprise object, holding it to perform certain actions. An active enterprise object carries a set of deontic tokens, which control the occurrence of *conditional actions* within its behaviour. These tokens are either permits, burdens or embargos. A deontic token is not itself an active enterprise object; it is held by exactly one active enterprise object.

Further, the constraint is expressed by a rule (see section II) forming part of the token; an appropriate notation for expressing this rule will be selected by the specifier. The notation allows the declaration of the active enterprise object and conditional action to which it applies, and requirements on other enterprise objects fulfilling roles in the controlled conditional action. For example, the rule may control the performance of a purchase action by a consumer and place restrictions on the supplier and the artefact being purchased. The notation may also declare periods of validity or deadlines for performance of the action. Allowable associated information will depend on whether the token is a permit, a burden or an embargo.

Burden is defined as a deontic token encapsulating the statement of an obligation on the active enterprise object holding it, thereby modifying the urgency of the active enterprise object in performing associated conditional actions within its behaviour.

Embargo is defined as a deontic token encapsulating the statement of a prohibition on the active enterprise object holding it, thereby modifying the ability of the active enterprise object to perform associated conditional actions within its behaviour.

Permit is defined as a deontic token encapsulating the statement of a permission on the active enterprise object holding it, thereby modifying the ability of the active enterprise object to perform associated conditional actions within its behaviour.

Conditional action is defined as an action, which has associated preconditions based on the sets of burdens, permits and embargos carried by the active enterprise objects filling its various action roles. The specification of the conditional action states what permits are required for, what burdens favour, and what embargos inhibit performance of the action.

RM-ODP EL makes use of the speech act theory, concerned with utterances that have performative function in language and communication [15]. Accordingly, *speech act* is defined as an action whose performance results in a change to the sets of deontic tokens (permits, embargos and burdens) carried by the active enterprise objects filling its various action roles. A speech act may result in the addition of new tokens to the performer of an action role, in the removal of tokens from the performer of an action role, or the transfer of tokens from the performer of one action role to the performer of another action role in the same interaction.

D. Accountability concepts

This section discusses how the deontic concepts can be used to support the expression of responsibility and accountability constraints. The aim is to support the traceability of obligations in the overlapping and interacting communities that make up the enterprise [11]. These concepts may be of increasing value when considering new directions in the accountable care organization (ACO) initiatives. The central notion here is that of *parties* (defined in IV.B) with broader responsibilities derived from some social or legal framework

Commitment is defined as an action resulting in an obligation by one or more of the participants in the act to comply with a rule or perform a contract. The enterprise objects participating in an action of commitment may be parties or agents acting on behalf of a party or parties. In the case of an action of commitment by an agent, the principal responsible for the agent becomes obligated. The fact that an enterprise object is obligated is expressed by associating with it a burden describing the obligation.

Prescription is as an action that establishes a rule.

Authorization is an action indicating that a particular behaviour shall not be prevented. Note that unlike a permission, an authorization is an empowerment. The fact that an enterprise object has performed an authorization is expressed by issuing a required permit and undertaking a burden describing its obligation to facilitate the behaviour.

Declaration is an action that establishes a state of affairs in the environment of the object making the declaration. The essence of a declaration is that, by virtue of the act of declaration itself and the authorization of the object making the declaration or its principal, the declaration action causes a state of affairs to come into existence outside that object.

Delegation is an action that assigns something, such as authorization, responsibility or provision of a service, to another object.

Evaluation: An action that assesses the value of something.

E. Policy concepts

Community is also a placeholder that defines constraints on the behaviour associated with the enterprise objects fulfilling roles in a community. These are typically described as rules, such as obligations, permissions and prohibitions.

For example, a rule may state that a person in a patient role in a hospital may inquire about progress of her treatment (permission), while another rule may dictate that a clinician in a an emergency physician role in an emergency department must examine the person in the patient role within 10 mins of arrival at the hospital (obligation); a further rule may say that, for privacy reasons, information about a patient must not be shared with any other person but a family member (prohibition).

F. Positioning FHIR concepts: business architecture

A business architecture for a specific digital health solution will typically define collaborative structures involved in delivery of healthcare using IT. This includes the definition of *parties* involved in particular clinical care provision, what policies apply to them when filling the corresponding *roles*, what is the *objective* of the clinical care delivery, what *actions* are performed by these parties, and how these actions cause *enterprise objects* to change state, e.g. create, update, etc. In RM-ODP, this is supported through specifying a *community contract*, or several community contracts if there are several related communities, which may be needed to form a *federation*. The FHIR resources such as person and organisation can, for example, provide an information representation of *parties*; the resources such as observation, procedure, diagnostic order and diagnostic report can represent information about clinical *actions*, while the medication statement, condition, familyHistory and procedure resources can represent *enterprise objects* in the community.

Community thus provides a context for clinical, administrative and IT interactions Depending on the application, it may be a simple and short lasting one, such as during one patient encounter with a clinician, e.g. examination, undertaking a blood test, measuring blood pressure and the subsequent payment to the administrative person. Even in these simple cases, the objective of this community is clear and responsibilities of individuals and organisations involved are well defined.

Although the utility of the community concept may not be obvious when used as context for describing FHIR resources for these simple interactions, its value is more evident when used for complex scenarios such as clinical workflows and care plans, as well as when supporting the description of policies that constrain interaction, such as consent and privacy. Further, concepts such as observation or procedure can be as simple or complex as required. Some real-world actions are not clearly delegated into one or the other. Hence a strong conceptual modelling foundation is essential to enable a consistent interpretation of community behaviour. The classic example is a procedure conducted for the intent of an observation. In this case the observation can't happen without the procedure and the procedure makes no sense except in light of the subsequent observation.

Further value can be delivered through the use of accountability concepts, which may be of relevance for many implementation guides that include policy, consent and accounatbility requirements. For example, one important FHIR development is how to model Privacy Consent Directives (PCD). This is a specification of a client's (e.g., patient, consumer) health information privacy policy, which expresses the constraints for granting or withholding authorization to collect, access, use, or disclose Individually Identifiable Health Information (IIHI) about the client. Effective PCD are a bilateral agreement between the client and an individual/organization in accord with law, regulation and organizational policies with regard to their content [14]. A simplified enterprise specification would include:

- the definition of a community contract identifying the community roles of client, organisation/individual and a Substitute Decision Maker who may act on behalf of that individual,
- the definition of the outer legal, regulatory and organizational policies that apply to this community,
- declaration, when a client authors/publishes their privacy preferences as a self-declared PCD,
- delegation policy according to which a healthcare client delegates authority to a Substitute Decision Maker who may act on behalf of that individual,
- specification of actions/process steps for activities such as Collection and Use, and/or Disclosure.

V. Information Viewpoint

An information specification defines *information objects* in a system, needed to support different concepts in the enterprise language, such as providing further detail of enterprise objects that are only identified in the enterprise specification. In addition, the information viewpoint specifies *information actions*, which describe the handling of information in the system. Recall that every action of interest for a modelling purpose is associated with at least one object (see III.A).

A. FHIR considerations

Most of the FHIR resources can be regarded as a special realisation of the ODP information objects. This includes the description of their content and also REST operations over the information objects. In addition, some clinical objects can be in different states and one then needs to identify information actions which contribute to state changes in a resource life cycle, e.g. status of the Encounter (planned | arrived | in-progress | finished | cancelled).

FHIR uses the foundation of an information object to harmonise many aspects of the FHIR specification. This ranges from static clinical content to infrastructural resources implying underpinning system behaviour. This simplification allows for a simple, common approach to FHIR use but also makes aspects of community behaviour and relationships inherent rather than explicit.

Consider for example FHIR CarePlan resources. At present this resource is represented as a single FHIR resource with a large number of elements that describe properties of care plans. Some of these are defined inline as part of the resource. Many would consider however care plan in terms of interactions between different parties and would expect to see these created separately rather than inline or otherwise more explicitly be bound to behaviour models rather than information associated with behaviour models. While the FHIR Care Plan can be treated as an instance of a single resource in memory, its specification can better support reuse if for example the concept of activity can be defined and managed independently.

The availability of a minimal set of ODP EL concepts, to support such a specification, and use of them by implementers, can outweigh the costs associated with emerging ambiguities as is increasingly evident from efforts to conceptualise workflow concepts in the FHIR community. Alternatively, there is a desire for ensuring that FHIR is simplified for the developer, providing a ubiquitous yet malleable foundation that does not burden early adoption with perceived rigour, and hence cost, of independent representations of such behaviour.

VI. Computational Viewpoint

The computational viewpoint specifies basic functionality of software applications, the services offered by *computational objects* (referred to as *technical* or *computational services*), independent of the details of the underlying distributed infrastructure on top of which the applications are running. In many respects this viewpoint encompasses the principles of SOA frameworks.

RM-ODP provides a rich framework for describing the bindings between computational objects that can be used to model complex connections between computational objects, such as ones that report exceptions to other parts of the system. Binding is a context created by establishing a communication relationship between two or more objects and a binding is created by a binding action [11]. Normally, a computational object initiates its interaction with another computational object by performing a binding action that allows them to connect to each other and start exchanging services and data. However, in some cases, particularly where multimedia communication is involved, one object can initiate a binding between a number of other objects; this is known as third-party binding [11].

A. FHIR considerations

Our observation is that it is often difficult in FHIR to provide the separation of information from behaviour related to each resource. Does an object imply behaviour or is it simply content derived from such behaviour being performed? This observation which is key for the computational objects but indeed propagates from the enterprise concerns in which one clearly needs to make the separation of resources into action versus enterprise object.

FHIR operations extend to request-reply and one-way notification patterns through construction of these mechanisms from more primitive REST operations and resource packages. A one-way notification is supported through a RESTful interaction to an endpoint with a structured information package (resource) through which the intention and meaning is conveyed. Within a resource like subscription, the request-reply paradigm is supported through providing a callback endpoint on the initial interaction. This may seem to be a rather labourious way of performing such actions and putting a burden on the parties to coordinate the two one-way interactions. It is none the less a legitimate instantiation of such actions.

These interaction types support a richer behaviour specification than the RESTful paradigm of managing state by Create/Read/Update/Delete actions on a set of identified resources. In fact, the recent FHIR addition does include a lightweight operation framework that seamlessly extends the RESTful API. In these cases operations are used (a) where the server needs to play an active role in formulating the content of the response, not merely return existing information, or (b) where the intended purpose is to cause side effects, e.g. the modification of existing resources, or creation of new resources.

Further, the Subscription resource provides a way to support a publish/subscribe pattern of interactions, which can be regarded as a simple binding between listeners and publishers, while supporting multiple notification channels, such as a websocket, email and sms. This is in line with the ODP computational binding object, which can be also used to support more complex interactions between multiple listeners and publishers, including support for multimedia streams.

Finally, specific HL7 work groups appear to be proposing computationally-oriented resources such as Action Definition (proposed by the Clinical Decision Support group), serving as a template for creating specific actions to be executed when certain triggers occur.

VII. Engineering and Technology Viewpoints

The engineering language includes concepts such as nodes, node structure and channels between nodes to specify the mechanisms to support distributed interactions between computational objects. The aim is to provide technology neutral infrastructure designs with much longer life than the technologies that support them (which will be listed in the technology viewpoint). This allows these technologies to evolve without invalidating the system designs, which are a company's major asset. For example, an electronic health record system can be realised using one or more nodes and a node can be either an in-house system or provided through private cloud technology. Note that many standards nowadays provide various mechanisms to support specific infrastructure components, such as Web Services, .NET, Java EE or REST.

The technology language provides concepts to specify the hardware and software products from which the system is built, to test that such an implementation complies with the specification as prescribed by the rest of the viewpoints (and to specify the plans and processes for their selection), as well as acquisition and evolution of the system parts (hardware and software products) during its lifetime.

The details of the engineering and technology specifications are beyond the scope of this paper.

A. FHIR considerations

FHIR makes use of standard integration infrastructure, such as Web-based standards, and associated protocols such as HTTP and URI for resource identification. It is interesting to note that some FHIR resources can be used to hide the details of the resource implementation and thus have a similar role to ODP transparencies identified in the engineering viewpoint. One example is the FHIR binary resource, introduced to support pure binary content using the same framework as other resources.

One can also define *conformance* points that can be used for the testing of FHIR resources against their specifications. There is obviously some similarity here with the intent of the FHIR implementation guides and we plan to explore this similarity in our future work.

VIII. Discussion

A. Current situation

The FHIR approach to building digital health applications is offering many new benefits to the implementation community, because of the use of web-based technologies and standards, adoption of RESTful framework, and freely available resource specifications. Further value comes from the community-based approach to testing and feedback and pro-active support by the core FHIR team. There is also no requirement for understanding the overall FHIR architecture and underpinning semantics of all resources, rather it is sufficient to select available resources and tailor them, through extensions or constraints, for the purpose of a specific application.

Architecture understanding and guidance is however becoming increasingly important in view of the proliferation of resources. This requires ensuring consistency in the use of resource elements, data types and terminologies across resources, and in the way resources are linked and composed. This in turn requires a strong architecture approach for describing resources and their relationships as well as a foundational clinical ontology base for the evolving resource set. The need for an architecture approach is further augmented by the need to better support behavioural aspects, i.e. services, workflow and policies that govern access to resources. The latter is evident in the Data Access Framework implementation guide and in supporting content governance through strong content provenance. However, the effort in documenting the architecture and roadmap is significant and requires an extra effort for the core team, restricting their ability to deliver expected FHIR features, in particular in view of the speed and agility of development.

Section II.B showed the taxonomy of FHIR resources based on the nature of the resources [4], rather than motivated by the need for architectural separation of concerns and use of separate modelling languages, as for example recommended by the RM-ODP. This is pragmatic for developers but creates challenges for those seeking an architectural view. Considering that the structure of resources and their use also depends on their architecture properties, e.g. whether they are focused on healthcare information artefacts or on behavioural artefacts, we believe the RM-ODP architecture viewpoints with their modelling languages can provide additional clarity and perhaps facilitate how resources and their elements are used as part of implementation guides and specific solutions..

B. Future considerations: adding behaviour semantics

The table below provides several examples for positioning FHIR Resources in the context of different RM-ODP behaviour concepts. Note that the third column indicates the ODP viewpoint abbreviation, 'E','I', 'C', 'N' and 'T' (for enterprise, information, computational, engineering and technology concepts), and 'F' for foundational concept. In some cases, we show in brackets when a foundational concept is used directly (without refinement) in a specific viewpoint. This is not meant to represent idempotent mappings but instead identify potential FHIR platform concepts that can be a target for the mapping from the ODP concepts. Note that some of the concepts in the table can have multiple mappings and that one can perform a similar analysis for other platforms, e.g. openEHR and CIMI.

ODP concept	FHIR concept (examples)	Vp
Community	Care Plan; Encounter	E
Community Role	Patient; Participant (in Encounter)	E
Enterprise Object (also Party)	Instances of Patient, Practitioner, RelatedPerson (supported by binding)	E
Action	Appointment	F
Action Template	Action Definition	F(C)
Behaviour	Action Definition (with relatedAction element)	F(C)
Event	Subscription; Message Header;	F(C)
Process	Workflow (not well defined)	F (E)
Step (in Process)	Task	F (E)
Policy (Obligation, permission, prohibition)	*No similar concept*	F
Federation	*No similar concept*	E
Business Service	HealthcareService; Location	E
Accountability	*No similar concept*	E
Location in space	Location	F
Location in time	*Not explicitly defined*	F
Information Object	Most Clinical Resources	I
Information Action	An action changing Encounter status (planned | arrived | in-progress | onleave | finished)	I
Composition (of objects)	Group; Bundle; Composition; Document	F
Computational Object	Service is most similar (yet not well defined) concept	C
Computational Interface	Operations	C
Computational Binding (primitive)	Subscription (channels: rest-hook | websocket | email | sms |)	C
Computational Binding (compound)	*No similar concept*	C
Engineering Object	*No similar concept*	N
Channel	*No similar concept*	N
Stub	*No similar concept*	N
Conformance	Conformance	T
Compliance	Profile; IG	T

IX. Conclusions and Future Directions

We presented an approach towards better specification of behavioural semantics for the needs of digital health enterprise. These semantics leverage the concepts from the RM-ODP standard, to ensure precision of expression, in a way that can accommodate a number of health informatics standards and infrastructures including distributed systems, cloud and mobile applications. We demonstrated through several examples how the emerging FHIR standard would benefit from explicit support for behavioural semantics and facilitate the development of health IT solutions [13].

In the near future we intend to extend the analysis of the initial semantic links between FHIR and RM-ODP concepts presented in section VIII. The aim would be to develop a more detailed mappings from the RM-ODP as a technology independent architecture framework and FHIR as a specific technology platform. This would allow us to use the comprehensive architecture framework of RM-ODP to look at the range of FHIR resources, possibly identifying where future FHIR work may be needed to build a more complete framework. In doing so, we intend to consult with the FHIR core team and get their input.

Acknowledgment

We would like to thank Grahame Grieve and the core team for their outstanding technical contribution, vision and efforts in the conceptualisation and driving of the FHIR agenda. We do think FHIR provides much promise for addressing challenges in the future of digital health including difficult interoperability problems.

References

[1] Health Level Seven International (HL7), hl7.org

[2] openehr.org/

[3] opencimi.org/about_cimi

[4] hl7.org/fhir/2016May/index.html

[5] Interoperability Framework V2.0, National e-health Transition Authority (NEHTA) digitalhealth.gov.au/implementation-resources/ehealth-foundations/interoperability-framework

[6] E-health Interoperabilty Framework, SA HB 137—2013, infostore.saiglobal.com/store/Details.aspx?ProductID=1635046

[7] HL7 Service-Aware Interoperability Framework (SAIF): Canonical Definition, Release 2, Nov 2014, hl7.org/implement/standards/product_brief.cfm?product_id=3

[8] Z. Milosevic, J.P Andrade Almeida, J. C. Nardi: Towards Better Semantics for Services in eHealth Standards: A Reference Ontology Approach, EDOC Workshops 2014: 276-285.

[9] ISO/IEC IS 15414, Information Technology — Open Distributed Processing — Enterprise Language, Third edition 2015-04-15.

[10] ITU-T/ISO, "ITU-T X.902 | ISO/IEC 10746-2, Information Technology Open Distributed Processing Reference Model – Foundations", 2010.

[11] P.F. Linington, Z. Milosevic, A. Tanaka and A. Vallecillo, *Building Enterprise Systems with ODP, An Introduction to Open Distributed Processing*, Chapman & Hall/CRC Press, 2011.

[12] ISO/IEC IS 19793, Information Technology — Open Distributed Processing — Use of UML for ODP System Specifications, 2008. Also pub- lished as ITU-T Recommendation X.906.

[13] A. Bond, A. Hacking, Z. Milosevic, A. Zander: *Specifying and building interoperable e-health systems: ODP benefits and lessons learned*, Computer Standards & Interfaces, special issue on Open Distributed Systems, April 2012

[14] http://hl7-fhir.github.io/pcd/pcd.html

[15] Speech act theory, https://en.wikipedia.org/wiki/Speech_act