Behaviour Models Clarify Definitions of Affordance and Capability

Ella Roubtsova
Open University of the Netherlands
6401DL,Valkenburgerweg 177
Heerlen, The Netherlands
ella.roubtsova@ou.nl

Vaughan Michell
Henley Business School
University of Reading
Whiteknight, Reading, RG6 6UD, UK
v.a.michell@henley.reading.ac.uk

ABSTRACT

There is ongoing work on conceptual modelling of such business notions as Affordance and Capability. We have found that such business notions as Affordance and Capability can be constructively defined using elements and properties of executable behaviour models. In this paper, we clarify the definitions of Affordance and Capability using Coloured Petri Nets and Protocol models. The illustrating case is the process of drug injection. We show that different behaviour modelling techniques provide different precision for definition of Affordance and Capability and clarify the conceptual models of these notions. We generalise that the behaviour models can be used to improve the precision of conceptual models.

Categories and Subject Descriptors

H.1.0 [**Information Systems**]: MODELS AND PRINCIPLES

General Terms

Design, Modelling

Keywords

Affordance; Capability; Goal; Conceptual Model; Behaviour Model; Validation

1. INTRODUCTION

The modelling community is universally agreed that conceptual models are sufficient to give the definition of any business notion. Although, it is true, as any artifact type has its metamodel, the useful or practical definition of some business notions demands a combination of business concepts and the elements of behaviour models. By the useful and practical, we mean that the definitions of business notions should indicate the use of these notions to control or improve the business.

BM-FA '14, July 22 2014, York, United Kingdom

http://dx.doi.org/10.1145/2630768.2630774.

In this paper, we investigate the role of behaviour modelling in understanding of notions of business affordance and capability. In section 2 we analyse the definitions given in literature. In section 3 we formulate the research question. In section 4 we try to clarify the hidden relations in those definitions using the behaviour models. In section 5 we identify the elements of behaviour mod els that complete constructive definitions of affordance and capability and propose a conceptual model for presentation of affordance and capability. Section 6 concludes the paper and identifies future work.

2. DEFINITIONS OF AFFORDANCE AND CAPABILITY IN RELATED WORK

There is a variety of definitions of two related notions: capability and affordance.

Capability can refer to the result of an action of an active agent [4], and to an object's ability [2] and the ability of groups of resources to perform a task via a process [10].

Gibson [5] defined affordance as "the property that the environment or physical system offered the animal to enable a possible useful transformation for the benefit of the animal".

Michell [14] defined capability as "a property of a resource (tangible or intangible) that has a potential for action or interaction that produces value for a customer via a transformation process that involves the interaction of the resource with other resources. At a detailed level the potential for action relates to affordance".

However, the affordance can only be realised by an animal or agent. Gibson [5] defined affordance as "the property that the environment or physical system offered the animal to enable a possible useful transformation for the benefit of the animal". Shaw [6] identified that environment ability or disposition, must be complimentary to the active agent (human, animal) disposition and ability. Shaw defined this ability of the animal to compliment the affordance properties of the object as "effectivity" i.e. the "capability of the animal" to use the object in a transformation. Wells [9] suggests effectivity relates to *the functional state of the animal* and its possible movements.

Turvey[19] defined affordance-effective duals showing opportunities for action depended on both animal and object properties interacting.

Ortmann and Kuhn [17] presented a conceptual model of affordance as an inheritor of qualities of entities and shown the relation between the affordance, actions and stimuli. They have given examples of calculation of affordance in Haskel (a functional language).

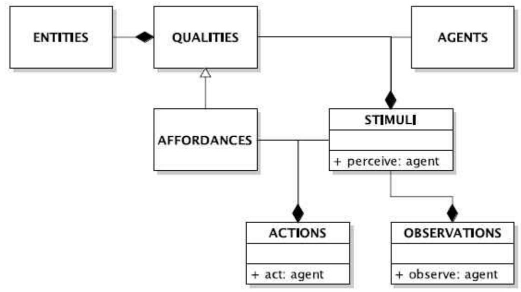

Figure 1: Conceptual Model of Affordance Ortmann and Kuhn [17]

Analysis of all these definitions shows that the affordance can be seen as a capability to fulfill an action or make a step (a transition) in a process.

However, the notion of business capability cannot be restricted to one step. Capability also refers to an objects ability [2] and the ability of groups of resources to perform a task via a process [10]. A capability results from interactions between two or more resources that achieve a business goal, typically to increase the business value of the transformed resources with respect to a business client.

A capability results from interactions between two or more resources that achieve a business goal, typically to increase the business value of the transformed resources with respect to a business client. Business capability is the potential to achieve a goal via an action or series of actions in a process resulting from the interaction of two or more resources, in a transformation that produces business value [14].

Kim [8] discussed how the affordance-effectivity dual can be decomposed into possible actions. The possible actions related to the affordances are actualised by an affordance mechanism acting in a defined space-time path [14]. The affordance mechanism represents the energy transformation that drives the transition. An affordance path is the set of possible space time arrangements that are necessary to enable the affordance mechanisms to act and execute the capability.

Warren [20] introduced the idea of specific values that characterised the conditions for affordance. The collective critical conditions [14] can be termed critical affordance factors that are needed to relate affordance and effectivity and formalise the mechanism of an action.

3. RESEARCH QUESTION AND RESEARCH METHOD

3.1 Research Question

Many concepts were named in the related work on affordance and capability.

The question is if these elements are sufficient to constructively define and calculate affordance and capability in any business case. Can we combine all the concepts that define affordance and capability together in a reusable conceptual model with clearly defined relations?

Such a conceptual model can be used for analysis of business models and making design and business decisions.

3.2 Research Method

Our method shows the role of behaviour modeling in clarification of conceptual models of complex business notions such as affordance and capabilities and their relations.

1)We have analysed executable behaviour models of several business systems. The models have been built in several behavioural semantics that support work with data [15]. The businesses belong to different domains. In this paper, we present the fragments of two models.

2)We have identified the concepts used for definition of affordance and capability in behaviour models. We identified the semantics of modelling elements representing the relations between the concepts used for definition of affordance and capability.

3)We propose a conceptual model of affordance and capability that can be reused in modelling and analysis of business systems.

4. AFFORDANCE AND CAPABILITY IN BEHAVIOUR MODELS

4.1 Capability and affordance with Coloured Petri Nets

4.1.1 Coloured Petri Nets

Coloured Petri Nets [7] possess all the expressive means needed for understanding and possible measurement of the notion of capability. Coloured Petri Nets (CPN) combine advantages of classical Petri Nets (PN) with the expressive power of complex data types "colors").

An initialized non-hierarchical CPN(net) without time stamps is a tuple [18]:

$$CPN = (C, B, V, P, T),$$

- C is a finite set of colors (data types), $c \in C$. Colors and variables are defined in declarations. We use data types and variables to represent resource types.
- P is a finite set of places, $p1, , pm \in P$, depicted by ellipses. Each place p possesses a bag bp.
- B is a bag of tokens (values) of colors $c \in C$ represented near places. We use tokens to model instances of resources.
- V is a set of variables of colors $c \in C$.

- T is a finite set of transitions depicted by boxes (Figure 5). A transition represents an action a and are denoted by verbs.
- Each transition is a tuple $t = (I, g, a, O)$:
  - I is a finite set of input arcs. An input arc is directed from a place p to transition t. An arc contains an expression of the color of place p.
  - g is a guard of transition t. Each guard is a Boolean function. By default each guard has value true. We use guards to model the affordance-effectivity dual.
  - a is a name of the action associated with the transition.
  - O is a finite set of output arcs. An output arc is directed from the transition to a place p. An arc is labelled with an expression. The type of the expression is the *Colset* of the output place.

A path from a given marking to a given goal marking in a CPN model is a sequence (can be both finite and infinite) of transitions

$$(I_1, g_1, a_1, O_1), ..., (I_n, g_n, a_n, O_n); n \in N.$$

Since every transition in a path has own input marking (pre-state) and output marking (post-state), a guard and an action (often called an event), it potentially changes the marking of the CPN model until the goal marking is reached.

4.1.2 CPN model of Drug Injection

Based on structured interviews conducted at a health trust hospital [14] and the injection best practice [1], we model the capability to inject a drug using a syringe [15].

Resources include an active resource a *nurse* and passive resources: a *syringe*, an *ampoule* containing a drug (eg Ketamine) and a *patient* (passive in this case).

The capability to "inject the drug" depends on a sequence of actions (path) with the correct disposition (state) of resources to inject the drug. The disposition (state) is expressed with places and guards in the CPN-model. If there is no a sequence of actions leading to the goal state, there will be no capability. If any actions do not have the correct conditions (presented with guards), i.e. any of the critical affordance factors and mechanism are incorrect, there will be no capability.

The key actions are, firstly, "the grasping of the empty syringe by the nurse" and then "pushing the plunger closed ready to draw up the drug". Then the nurse places the syringe in a drug container (ampoule) and pulls the plunger to draw up the drug. The nurse holds the syringe in a different way - "hold to inject" and looks for a vein on the patient. Having perceived the vein the nurse pushes the syringe into the vein at the correct position and angle and then presses the syringe plunger to inject the drug.

Decomposing this process sequence into actions (CPN transitions labelled T) and situations (places labelled P) enables us to identify the critical state transitions and affordances/effectivities. Figure 2 shows a CPN model of the capability to "inject the drug". Figure 3 describes the transitions [15].

The initial state and the goal state of the business process are modelled by places that may contain tokens of given colors. Places are connected via transitions so paths leading from initial states to goal states relate to the capability of the system, i.e. the CPN simulation reaching the goal state. Tokens - represent instances of business object and agent actions and values for the dispositions of each resource (object or agent) at the point of interaction. Transitions represent the transformation affordance-effectivity interactions. A transition T of a CP-net is enabled if places of all its input arcs contain tokens to give values to input expressions of T, and the guard value is true. The guard values represent the critical affordance factors. For example, in *T7*, the "nurse perceives vein", must be true for injection to occur. Each enabled transition t can fire. When a transition t fires then for each input arc its expression is evaluated by a token from the arc's place. For each output arc its function is calculated using the values of the variables from the input arcs of the transition. The result of the output function is added as a token into the place of the output arc.

The types of *Resources* are represented as colsets shown below.

```
Declarations:
      colset Unit=unit;
      colset Drug=string;
      var d: Drug;
      colset ForcePlungerClosed=int;
      colset NurseName=string;
      colset SyringeName=string;
      colset Quality=bool;
      colset ForceHand=int;
      colset ForceSlip = int;
      colset ForcePlunger = int;
      colset ForceCrush = int;
      colset SyringeScale = int;
      var fh:ForceHand;
      var fs:ForceSlip;
      var fsp:ForcePlunger;
      var fc:ForceCrush;
      var qn:Quality;
      var s:SyringeName;
      var n:NurseName;
      colset SyringeopenScale= int;
      var Ls:SyringeScale;
      var Lco:SyringeopenScale;
      colset GripPattern= with grasp | hold;
      var gp:GripPattern;
      colset PerceiveVein=with perceives | notperceives;
      var pv:PerceiveVein;
      colset Nurse =product NurseName*Quality* ForcePlunger*
      ForceHand*GripPattern*PerceiveVein;
      colset Syringe =product SyringeName*ForcePlunger*
      ForceSlip*SyringeScale*SyringeopenScale;
      colset NSGrasped = product Nurse*Syringe;
      colset Container=string;
      var c:Container;
      colset ContainerDrug = product Container*Drug;
      colset NSReadyDrugDraw = product NSGrasped*ContainerDrug;
      colset NSdloaded = product NSGrasped*Drug;
      colset Patient=string;
      var pat:Patient;
      colset NSInjectPatient = product NSdloaded*Patient;
      val ForcePlunger=3;
      colset PatientInjected = product Patient*Drug;
```

4.1.3 Concepts related to affordance and Capability in the CPN model

Qualities of active resources (agents) and passive resources (entities) are represented by attributes (properties) of resources. E.g. the *Syringe* attributes are *name, slip force, crush force, syringe scale level, scale zero*. The forces are part of the necessary affordance mechanism that enables the syringe to be constrained. A token presents a syringe and contains values of variables: s (name), fs (slip force), fc (crush force), Ls (syringe scale level), Lco (scale zero)).

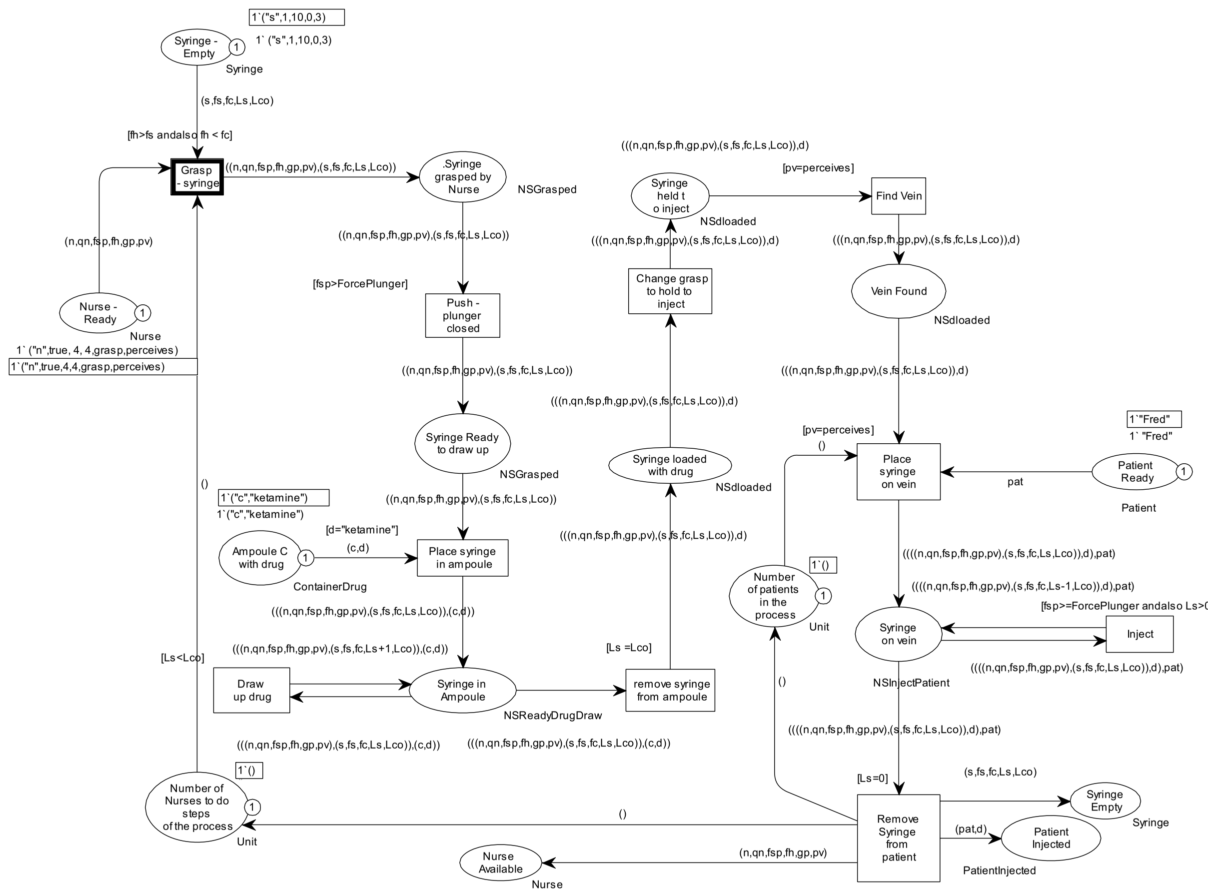

Figure 2: CPN model of the Syringe Injection

Ref	CPN TERMS: P =place, T = transition	PATH (at point of affordance-effectivity)	MECHANISM	CRITICAL AFFORDANCE FACTORS (path/mechanism)	RESOURCE CONDITIONS
P1	Syringe			fs=1N, fc = 10N,	Empty/clean
T1	**GRASP SYRINGE:** The empty syringe is grasped without slipping, then pushed closed to draw up the drug. The The critical affordance expressions are shown by the guard conditions on 'grasp syringe'. hands fit round syringe, grasp force fs must be greater than slip force (1N) but less than crush force (10N).	hands fit round syringe	grasp forces fh	fh > fs, fh< fc, hands fit round syringe, grasp force fs must be greater than slip force (1 N) < crush force (10N)	Syringe s slip and crush forces, scale ls = 0 and lco =3cm
P2	Nurse				Nurse
P3	Syringe grasped by nurse				NSGrasped - the nurse grasping the syringe
T2	**PUSH PLUNGER CLOSED:** To draw up the drug the syringe plunger is pressed closed by the nurse applying a force fsp > a minimum plunger force. Otherwise the drug cannot enter the syringe.	hand attached to plunger, plunger at end of syringe ls = lsc = 0	plunger force + fsp	fsp1 > forceplunger (min force to move it)	Syringe held in closed position
P4	Syringe ready to draw up (plunger in closed position)			ls =0	NSGrasped - with
P5	Ampoule C with drug			drug type - ketamine	ContainerDrug - an
T3	**PLACE SYRINGE IN AMPOULE:** containing the correct drug. If not in drug no drug will be drawn up (capability failure). The mechanism here is the gasp force holding the syringe and the ampoule - not shown	syringe needle immersed in drug	nurse grasp forces on syringe & ampoule	d=ketamine	ketamine is the correct drug/label for patient
P6	Syringe in Ampoule			ls = 0	NSReadyDrugDraw
T4	**DRAW UP DRUG:** Plunger is pulled back to draw up drug to the correct amount in increments of ls + 1 . Mechanism here is pulling force on the plunger creating a partial vacuum and atmospheric pressure forces the drug into the syringe.	Hand on plunger moved to end of syringe ie ls = Lco = 3	negative plunger force = -fsp	fsp1 > forceplunger (min force to move it), ls = 3	-fsp applied (not shown) ls = ls +1 until ls = 3 on scale
P6'	Syringe in Ampoule			ls = 3	NSReadyDrugDraw
T5	**REMOVE SYRINGE FROM AMPOULE:** The draw up drug continues until ls = 3.= Lco. Incorrect amounts = capability failure / patient not anaesthetised			ls = 3	syringe not in ampoule
P7	Syringe loaded with drug			ls =Lco	NSdloaded
T6	**CHANGE GRASP TO HOLD TO INJECT:** Nurse's finger tip locations/forces adjusted for safe drug injection at correct angle. Failure risks patient injury and not/partly injecting the drug	Grip pattern (position of fingers) = hold	grasp forces fh	fh > fs, fh< fc, GripPattern = hold	Syringe constrained in'hold to inject' position with no slip
P8	Syringe Held to inject			gp = hold	NSdloaded
T7	**FIND VEIN:** A vein on the patient is perceived, based on the nurse' knowledge. Mechanism is nurse' perception/cognition, visual ability (No vein, incorrect site = capability failure)	Visual path: ie nurse can see the patient and the site of injection	perception-cognition mechanism (Pv)	Pv = true	Nurse - has updated knowledge - Vein is found
P9	Vein Found			pv = perceives	NSdloaded
P10	Patient Ready			pat = Fred	Nurse sees the vein
T8	**PUSH SYRINGE IN VEIN:** at correct angle and position			pv = perceives,pat = fred	Correct patient vs drug
P11	Syringe in Vein			ls >0	NSInjectpatient
T9	**INJECT:** Plunger pushed closed at correct injection site to ensure drug is transferred to the patient, (otherwise no anaesthesia and capability fails)	syringe plunger location ls = 0	fsp	fsp > min, ls > 0	Syringe held in closed position
P11'	Syringe in Vein			ls = 0	NSInjectpatient
T10	**REMOVE SYRINGE...** - Inject contintues until ls = 0 (syringe can be withdrawn)			ls= 0	Syringe not in patient
P12	Syringe Empty			ls= 0	Syringe
P13	Nurse Available				Nurse
P14	Patient Injected				Patient Injected

Figure 3: Description of model Elements

A *Nurse* has properties: *name, quality, plunger force, hand force, grasp type, perceives.* The properties are needed to express the affordance mechanism in terms of the forces, that the nurse must exert, and the knowledge variables, such as the type of grip needed. The properties are implied by the token of type *Nurse*: (n (name), qn (quality), fsp (plunger force),fh (hand force), gp (grasp type), pv (perceives)).

Affordances are modeled as guards of transitions. An *affordance* is a boolean function that is calculated using attributes of tokens consumed by the fired transitions and constants representing environment. For example, the guard of the transition *Push Plunger Closed*

$$fsp > Force\ Plunger$$

relates the minimum force needed to move a plunger $Force\ Plunger$ and the force applying by the nurse fsp.

The Goal State in CPN is a goal marking, i.e. a vector of places (ellipses) containing tokens. The final goal state of the process is *(Patient Injected, Syringe is Empty, Nurse Available, Number of Nurses in the process, Number of Patients in the process).* The goal presentation is not elegant as place *Patient Injected* specifies the goal, but other places are related to the "process organization". The combination of places of different objects and concerns in one marking is the feature of the CPN semantics (and of the Petri Nets in general).

Affordance Path [14] is a sequence of transitions from the initial marking to the goal marking. Each transition of this sequences has own affordance (guard) depending on the values of tokens enabling transitions.

Capability.
What is the *Capability* in the CPN model?

Capability Minimum. We define *Capability Minimum* as the existence of a path from the initial marking to the goal marking and the existence of the tokens (with the right values of attributes) enabling the transitions on this path.

Capability Minimum is a boolean function calculated via model analysis (reachability of the goal marking):

```
if  there is a Path of the Process
    (including the affordance mechanism)
        from the initial state(marking)
        to the goal state (marking)
            then Capability Minimum = true;
            otherwise Capability Minimum = false;
```

Capability Maximum. We define *Capability Maximum* as a number of paths from the initial marking to the goal marking and the existence of the tokens (with the right values of attributes) enabling the transitions on these paths.

Capability Maximum is an integer function calculated via finding and summation of all paths from the initial marking to the goal marking. *Capability maximum* can be modified by changes the numbers of resources (tokens in places *Syringe, Nurse, Ampoule*, values of attributes of resources and by changing the process.

```
Counter=Number of all Paths of the Process
    from the initial state (marking) to the goal state (marking);
Capability Maximum= Counter;
```

Both definitions are useful for businesses. The *Capability Minimum* is useful for design decisions and planning of improvements to a business process. The *Capability Maximum* allows comparison of business processes.

The syringe case study demonstrates only *Capability Minimum.* There is a path from the initial state to the final state: *Grasp syringe, Push plunger closed, Place syringe in ampoule, Draw up drug (pull n times), Remove syringe from ampoule, Change grasp to hold to inject, Find vein,Place syringe on vein, Inject (Push m times), Remove syringe from patient.* The initial marking is: *(Patient Ready(1 token), Syringe is Empty (1 token), Nurse Available (1 token), Number of Nurses in the process (1 token), Number of Patients in the process (1 token)).* The goal marking is: *(Patient Injected(1 token), Syringe is Empty (1 token), Nurse Available (1 token), Number of Nurses in the process (1 token), Number of Patients in the process (1 token)).*

By the changes of the number of resources, the properties of recourses and/or the changes of the process, the number of paths leading to the goal state can be increased. In future work we plan to demonstrate the use of the notion of *Capability Maximum.*

4.2 Capability and Affordance in Protocol Models

In this section we show that another behaviour modelling semantics called Protocol Modelling reveals a new element of the affordance notion.

Protocol Modelling. The building blocks of a Protocol Model [13] are protocol machines and events. They are instances of correspondingly *protocol machine types* and *event types.*

A protocol machine type is an LTS extended to enable modelling with data:

$$PM_i = (s_i^0, S_i, E_i, A_i, CB_i, T_i),\ where$$

- s_i^0 is the initial state;
- S_i is a non-empty finite set of states.
- E_i is a finite set of recognized event types e_i, coming from the environment.
- A_i is a finite set of attributes of different types. The set can be empty.
- $CB_i(PM_1, ..., PM_n, E_1, ..., E_m) = (PM_1, ..., PM_n, E_1, ..., E_m)$
is a callback function for updating the values of the attributes, states and events of the protocol machines of the protocol model. $PM_1, ..., PM_n$ are the protocol machines of the protocol model. $E_1, ..., E_m$ are events of the protocol model.
- $T_i \subseteq S_i \times E_i \times S_i$ a finite set of transitions:
$t = (s_x, e, s_y),\ \ s_x, s_y \in S_i,\ e \in E_i$. The set of transitions can be empty. The states may be updated without callback functions. The values of the attributes, states and events may be updated using the callback function only as a result of a transition, i.e. as a result of an event acceptance.

In order to facilitate reuse, protocol machines come in two variants: Objects and Behaviours. Behaviours cannot be instantiated on their own, but may extend functionality of objects. In a sense, objects with included behaviours are similar to mixins or aspects in programming languages [3; 11].

An event type is a tuple $e = (EventName, A^e, CB^e)$ where.

- A^e is a finite set of attributes of the event.
- $CB^e(PM_1, ..., PM_n, E_1, ..., E_m) = (PM_1, ..., PM_n, E_1, ..., E_m)$ is a callback function corresponding to this event. The callback function for an event is used if the event calculates attributes of generates other events from the state of the model.

Within the Protocol Modelling, the callback functions are the instruments for data handling. In the ModelScope tool [12] supporting execution of protocol models, the callbacks are coded as small Java classes with methods changing and/or returning the values of attributes and states of instances of protocol machines. They may also change attributes of events and generate event instances.

CSP parallel composition. In any state, a system model PM is a CSP parallel composition of finite set of *instances of protocol machines*.

$$PM = \underset{i=1}{\overset{n}{\|}} PM_i = (s_0, S, E, A, CB, T), \quad n \in N.$$

A Protocol Model PM is also a protocol machine, the set of states of which is the Cartesian product of states of all composed protocol machines [13]:

$s_0 = \prod_{i=1}^{n} s_i^0$ is the initial state;

$S = \prod_{i=1}^{n} S_i$ is the set of states;

$E = \bigcup_{i=1}^{n} E_i$ is the set of events;

$A = \bigcup_{i=1}^{n} A_i$ is the set attributes of all machines;

$CB = \bigcup_{i=1}^{n} CB^i$ is the set of callbacks of all machines.

Dependent Protocol Machines. Derived states. Usually transitions T_i of a protocol machine PM_i enable updates of its own states. On the other hand, protocol machines can read the states of other protocol machines, although cannot change them. This property makes possible the dependency of protocol machines. A dependency means that one protocol machine needs to read the state of another machine to calculate its own state and/or the attributes. Such calculated states are called *derived states*, which distinguishes them from the protocol machine states denoted in the model, which are called *stored states* [13]. Callback functions CB_i are used to update attributes and calculate derived states.

Two possibilities are used in dependent machines:
(1) The pre-state of a transition can be calculated. The pre-state is similar to guards calculated in CPN [7] and the UML state machines [16].
(2) The predicted post-state of a transition can be calculated and used to *allow* or *refuse* the event.

In the CPN model we modelled affordances as guards. Guards are calculated using the values of variables in the pre-state of the model (marking). Protocol models also enables calculation of affordances from pre-states in protocol machines with derived states. *In addition to this, the semantics of prediction of the post-state in Protocol Modelling reveals the additional semantics for affordance, namely, its dependence from the post-states.*

4.3 A Fragment of the Protocol Model of Drug Injection illustrating the possible relation of affordance with post states

Let us imagine a syringe with automatic drawing up of drugs. The drug may be very aggressive. Overdosing may cause health problems.

The allowed volume of the drug may be set to the syringe as a constant Max. The controlling device of the syringe prevents drawing up above the given *Max* volume. We oversimplify the model and do not show the relation between the force and the drawn up volume. The complete Protocol Model of the Injection is the future work.

However, the presented fragment illustrates the possible relation of the affordance of the step *DrawUpDrug* and the post-state of this step. The post-state depends on the value of the variable *Volume* and is expressed as derived state *"within max"* of the behaviour *Volume Control*. If the state *"within max"* is predicted after acceptance of event *DrawUpDrug*, the event is enabled and *affordance= true*. If the state *"over max"* is predicted, then *affordance=false* and the event is refused.

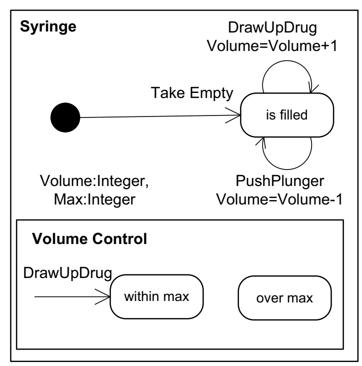

Figure 4: A fragment of the Protocol Model of Syringe: Draw up drugs with volume control

```
MODEL Syringe3
# OBJECT definitions

OBJECT Syringe
    NAME Syringe Name
    INCLUDES Volume Control
    ATTRIBUTES Syringe Name:String, Volume:Integer,Max:Integer
    STATES   is filled
    TRANSITIONS @new*TakeEmpty=is filled,
                is filled*!DrawUpDrug=is filled,
                is filled*!PushPlunger=is filled,

BEHAVIOUR !Volume Control
    STATES within max, over max
    TRANSITIONS @any*DrawUpDrug=within max

# EVENT definitions

EVENT TakeEmpty
```

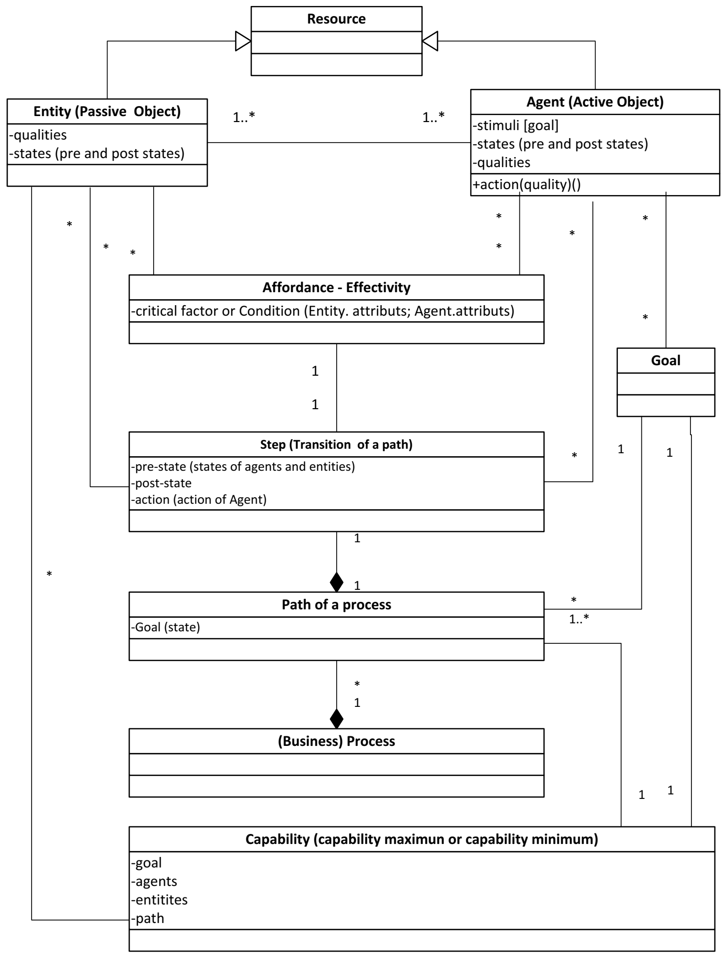

Figure 5: Conceptual Model of notions Affordance and Capability

```
ATTRIBUTES
    Syringe:Syringe,
    Syringe Name:String,
    Max:Integer

EVENT DrawUpDrug
ATTRIBUTES
    Syringe:Syringe,

EVENT PushPlunger
ATTRIBUTES
    Syringe:Syringe,
-----------

package Syringe3;
import com.metamaxim.modelscope.callbacks.*;

public class Syringe extends Behaviour {
public void processDrawUpDrug(Event event, String subscript){
int newVolume = this.getInteger("Volume") + 1;
this.setInteger("Volume", newVolume);
}

public void processPushPlunger(Event event, String subscript){
int newVolume = this.getInteger("Balance") - 1;
this.setInteger("Volume", newVolume);
}
}

public class VolumeControl extends Behaviour {
public String getState() {
if (this.getInteger("Volume")>this.getInteger("Max"))
            return "over max";
               else return "within max";
}
}
```

The dependence of the affordance from the post-state of the corresponding transition does not contradict any previous definitions that state that affordance is related to the step (a transition) of a business process. Indeed, a transition is a tuple *(pre-state, action, post-state)* [16]. If affordance is related to a transition, it may be related to any element of the tuple: pre-state, action and post-state.

5. CONCEPTUAL MODEL OF CAPABILITY AND AFFORDANCE

Figure 5 presents our conceptual model of affordance and capability. It shows the relations of concepts that we found in existing definitions and identified in behaviour models.

The conceptual model shows that affordance (shown in Figure 5 as affordance-effectivity to confirm the terminology used by other authors) is a function that belongs to a step (a transition). Affordance is derived from the attributes and actions of agents (active resources) and entities (passive resources) that communicate to achieve a goal (system state).

A path of a business process is a set of steps (transitions) with their affordance. We are interested in paths that lead to the goal state. The paths are composed into business processes. The capability is calculated as the existence of at least one path leading to the goal state or as a number of such paths. There is one capability minimum and one capability maximum function for each goal.

As always with business models, the difficulties of conceptual modelling are caused by the different terminology used by the business community and modelling community. The behaviour models clarify the semantics of the terminology and form the basis for mutual understanding.

6. CONCLUSION AND FUTURE WORK

The contribution of this paper is twofold:

- Firstly, the conceptual capability-affordance model has been constructed on the basis of analysis of behaviour models. The behaviour models clarify the notions of affordance and capability to the constructive definitions. These notions may be used for comparison of businesses and services. Indeed, a service that has more paths to achieve the goal state, than another, is more capable of achieving the goal and may attract more users.

- Secondly, the paper makes a first step to a method for clarification of conceptual models using different executable behaviour modelling semantics. Different executable modelling semantics clarify the relations between concepts and can be productively used as validation instruments for conceptual modelling.

In future work we are going to validate our conceptual model for capability and affordance with logical methods.

Another direction of our future work is developing the systematic procedures for conceptual modelling clarified by the behaviour models. Many business concepts are derived not only from other concepts or their attributes, but also from the analysis of behaviour models, both static analysis and the simulation results. Developing the systematic procedures for clarification of conceptual models with the behaviour models can improve the quality of conceptualization. In future work, we will focus on development of such systematic procedures.

References

[1] A. Andre. What Makes A Prefilled Syringe Usable And Ergonomic. *http://www.usernomics.com/AnthonyAndre. PDApresentation.WhatMakesAPrefilledSyringe UsableAndErgonomic.July2011.pdf*, 2011.

[2] D. Beimborn, S. Martin, and U. Homann. Capability Oriented Modelling of the Firm. *IPSI Conference*, 2005.

[3] G. Bracha and W. Cook. Mixin-based inheritance. *OOPSLA/ECOOP '90 Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications*, pages 303–311, 1990.

[4] F. Gallouj and O. Weinstein. Innovation in services. *Research Policy*, 26(4-5), 1997.

[5] J. Gibson. The Ecological Approach to Visual Perception. *Houghton Mifflin Company, Boston, California Management Review*, 1979.

[6] J. Greeno. Gibson's affordances. *Psychological Review*, 101(2):336–342, 1994.

[7] K. Jensen. *Coloured Petri Nets*. Springer, 1997.

[8] N. Kim, H. Thiruvengada, L. Rothrock, R. Wysk, and D. Shin. Modelling of affordances in human-involved complex systems using finite state automata (FSA). 2008.

[9] A. Lenarcic and M. Winter. Affordances in Situation Theory. 25(2):155–181, 2013.

[10] R. Makadok. Towards a synthesis of the resource based dynamic capability views of rent creation. *Strategic Management Journal*, 22:387–401, 2001.

[11] A. McNeile and E. Roubtsova. Aspect-Oriented Development Using Protocol Modeling. *LNCS 6210*, pages 115–150, 2010.

[12] A. McNeile and N. Simons. http://www.metamaxim.-com/.

[13] A. McNeile and N. Simons. Protocol Modelling. A Modelling Approach that Supports Reusable Behavioural Abstractions. *Software and System Modeling*, 5(1):91–107, 2006.

[14] V. Michell. The Capability Affordance Model: Comparing Medical Capabilities. *LNBIP 109*, 2012.

[15] V. Michell and E. Roubtsova. Modelling Capability and Affordance as Properties of Human/Machine Resource Systems. *Accepted for publication in the Proceedings of the 4h International Symposium on Business Modeling and Software Design, BMSD 2014, Luxembourg*, 2014.

[16] OMG. *Unified Modeling Language: Superstructure version 2.1.1 formal/2007-02-03*. 2003.

[17] J. Ortmann and W. Kuhn. Affordances as Qualities. *Formal Ontology in Information Systems Proceedings of the Sixth International Conference (FOIS 2010).IOS Press. Frontiers in ArtificialIntelligence and Applications*, 209:117Ű130, 2005.

[18] E. Roubtsova. Property specification for Coloured Petri Nets. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC2004), Editors W.Thissen, P.Wieringa, M.Pantic, M.Ludema, IEEE Catalog Number 04CH37583C, ISBN 0-7803-8567-5, October 10-13 2004, The Hague, The Netherlands*, 2004.

[19] M. Turvey. Affordances and Prospective Control: An Outline of the Ontology. *Ecological Psychology*, 4(3):173Ű187, 1992.

[20] W. Warren. Perceiving Affordances: A Visual Guidance of Stair Climbing. *Journal of Experimental Psychology: Human Perception and Performance*, 10(5):683–703, 1984.