White Paper

Agent-Oriented BPMN

WP0104 | September 2013

Bhakti Stephan Onggo

Bhakti Stephan Onggo is a lecturer at the Department of Management Science at the Lancaster University Management School, United Kingdom. His research interests are in the areas of simulation methodology (modeling paradigms and conceptual modeling), simulation technology (parallel and distributed simulation, cloud-based simulation) and business process modeling and simulation applications. He has carried out a number of consultation projects in healthcare, manufacturing and public sector services including the European Commission and a UK Police department.

In a process-oriented modeling view, we usually see the world as a collection of interacting activities in which entities move from one activity to another. For example, in a manufacturing system, we often view it as a collection of related manufacturing activities (usually conducted at various machines) in which raw materials move from one activity to another and are transformed into various work-in-progress items until they leave the system as final products (or rejects). While this view is common, there are cases where an organization would prefer to be perceived as providing human-centered services rather than process-centered services. This is especially true for (although not limited to) organizations offering services to their customers. The objective of this white paper is to introduce the concept of agent-oriented modeling (or widely known as agent-based modeling in academic literature).

This white paper will also discuss how a business process model is represented from the agent-oriented perspective. The modeling language that will be used in this white paper is BPMN. BPMN is a modeling language and standard controlled by the Object Management Group (OMG). BPMN is designed to support a process-oriented modeling view. Hence, at the first sight, BPMN does not look like to be the right choice to represent an agent-oriented model. This white paper will explain how BPMN can be used to represent a model that emphasizes more on the agents who are involved in a business process. We use the word "agent" to make it more universal covering human agents (people), social/organizational agents consisting of human (such as departments and sales team) and non-human agents (such as machines). The advantage of using BPMN is that we do not have to invent a new modeling language and we can use the modeling power offered by BPMN.

Agent-oriented modeling

I believe the main challenge of using agent-oriented modeling to represent a business process is not on the technique but on the change of mindset, that is, on how we change our view about the world. Instead of seeing the world as a collection of interacting activities, we see the world as a collection of interacting agents. Instead of focusing on processes and treat agents as secondary, we focus on agents and processes are formed by the interactions between agents. Since in most cases, agents include people and social/organizational constructs comprises people, we put people at the center of our model. This will make us more aware of the objectives and concerns of people who are requiring or providing services in the system being modeled such as the customers and the key human assets.

What is agent-oriented modeling?

An agent-oriented model can be defined as a model that is formed by a set of autonomous agents that interact with their environment and other agents through a set of internal rules to achieve their objectives. Hence, the starting point in agent-oriented modeling is the identification of agent types in our model and their interactions.

An agent is an entity who can make an independent decision in order to achieve certain objectives. Each agent has a set of characteristics (or attributes) and is able to perform activities (or behaviors). An agent type is a modeling construct that represents all agents that can be identifiable by using the same set of characteristics and are able to perform the same set of activities. In a health clinic, the types of agent include patient, clerk and doctor. A patient is an agent type because all patients can be identified using the same set of characteristics (such as patient id/social security number, name and severity level) and are able to perform the same activities (such as arrive at the clinic and filling a form). Of course, each patient has a unique identification number. Likewise, some patients may need more time to fill the form. In agent-oriented modeling, it is important to understand the difference between agent and agent type. An agent is an instance of agent type, i.e., from one agent type we can generate one or more agents of the same type.

The second step is to identify the relevant interactions between different agent types. In our health clinic example, a patient interacts with a clerk in a health clinic when the patient registers and makes payment. A patient interacts with a doctor during the consultation. A clerk interacts with a doctor by telling the doctor that a patient has registered and is waiting for the doctor. Figure 1 shows how our simple agent-oriented model looks. In this example, when a patient arrives at a health clinic, the patient needs to register with the clerk. The registration will put the patient in a queue to see a doctor. A doctor will see the patient when the doctor is ready. When the consultation completes, a signal is raised to tell the clerk and the patient that the consultation is complete. The clerk and the patient will interact in the payment activity. This example serves as an illustration only so we do not show all activities such as when the clerk and the doctors arrive in the morning.

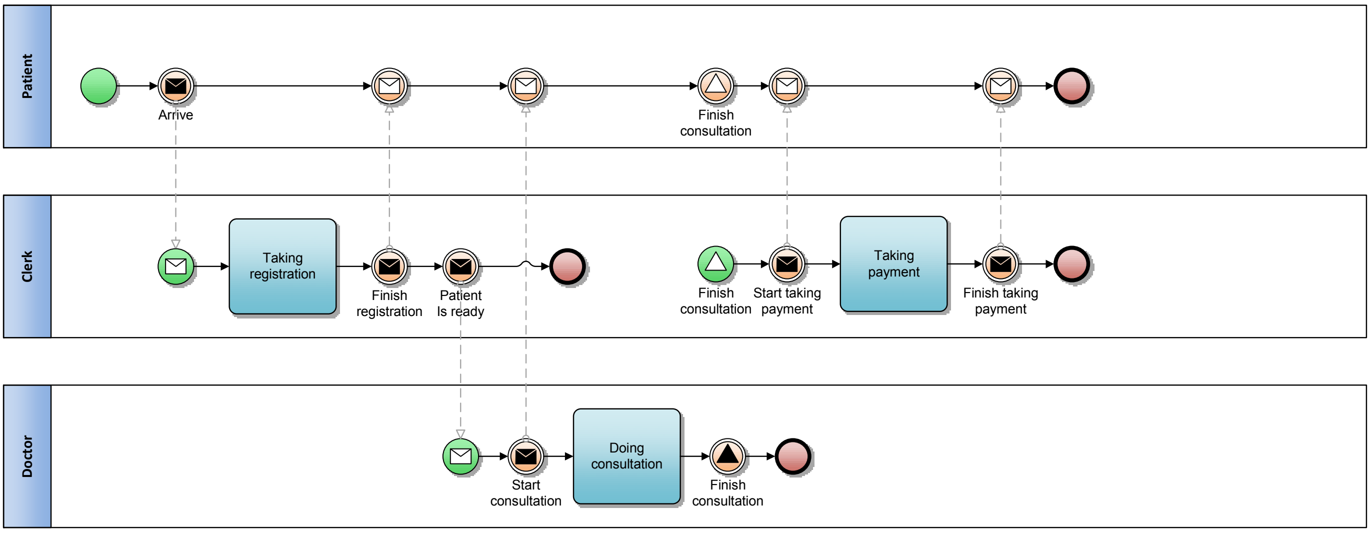

Figure 1: Agent-oriented model of a simple health clinic

As in any modeling work, in agent-oriented modeling, we need to define the model boundary which includes a decision on which agent types will be included in the model. The boundary also includes the characteristics, behaviors and interactions that we think are relevant. The decision on model boundary should be based on the objective of the modeling work.

Comparison with process-oriented modeling

If we compare the agent-oriented approach with the process-oriented approach, in a process-oriented approach, we often start with the identification of processes. In our health clinic example, there are three processes: registration, consultation and payment. Figure 2 shows a typical model for the same health clinic but the model is represented using a process-oriented approach. In this example, the process-oriented model views a patient as an outsider (drawn as a black box). This model may give an impression that we pay more attention to the processes than the patient (or any customer in general). The process view can also make us miss the fact that some of the processes are done by the same agent (for example, the registration and payment are done by the same clerk). Hence, we may overlook some resource contention problems in the system and other resource related issues such as imposing inappropriate performance indicators and allocating imbalanced workload. The inability to identify agents acting on multiple roles is likely to pose a problem for larger and more complex systems in which agents with multiple roles are common. This is not to say that a process-oriented model is inferior to an agent-oriented model because the choice should really depend on the objective of the modeling work and the nature of the business processes being modeled.

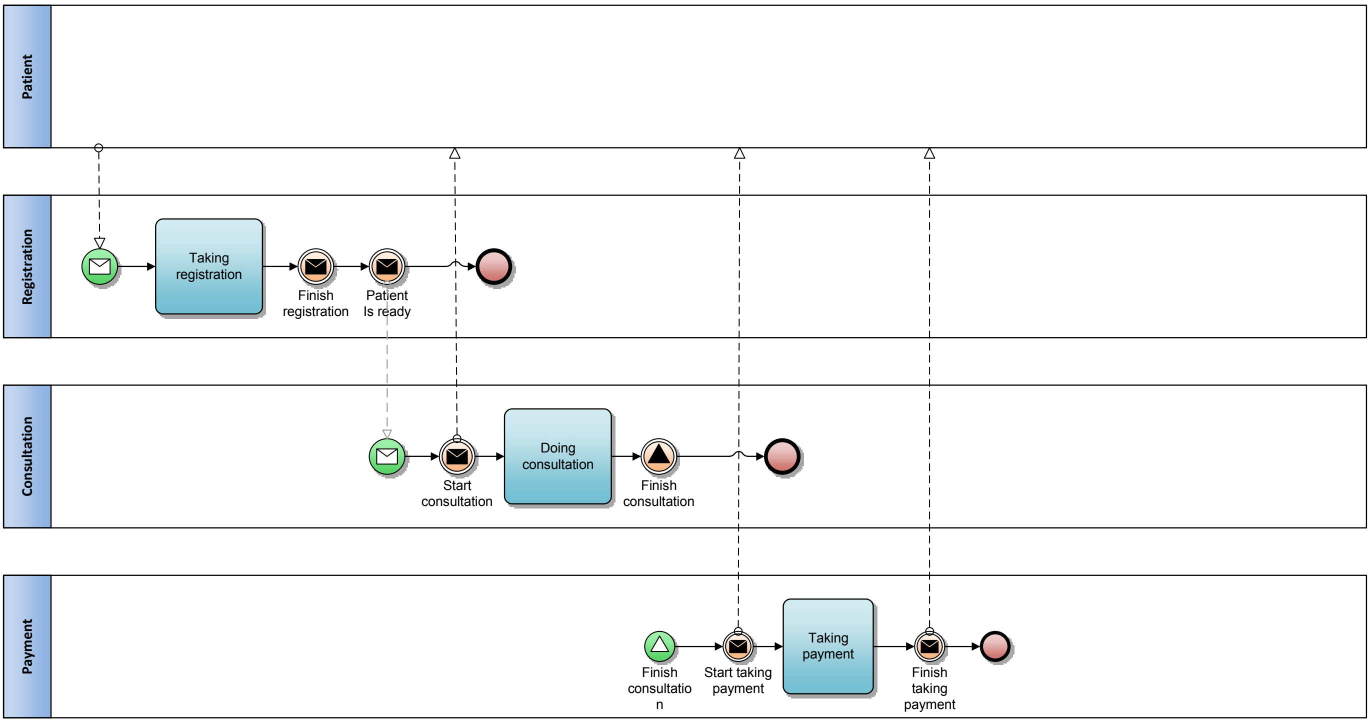

Figure 2: Process-oriented model of the same health clinic

Agent-oriented modeling using BPMN

Advantages

The use of BPMN for agent-oriented modeling has a number of advantages. First, BPMN is a widely known standard designed for business users. Hence, BPMN can be accepted more readily by business users in comparison to creating a new modeling language for agent-oriented modeling. The second advantage is that the BPMN standard is supported globally across various organizations. Hence, tools that support BPMN have become widely available including iServer Process Modeler, which also provides a central storage point for the processes themselves. Furthermore, the use of BPMN does not require any license fee. Finally, BPMN comprises two parts: a business view and a technical view. The advantage of having these two parts is that we can use BPMN for different purposes and for different audiences. The business view targets the business analysts who will use BPMN to create descriptive business process models which can be communicated and analyzed. The technical view targets technical developers who will need to add detailed technical specifications to the models to make them executable. The technical view should allow us to use computer simulation to analyze the dynamic behavior of our model. Computer simulation is very useful for process analysis and design. Despite these advantages, we need to demonstrate that BPMN meets the requirement for an agent-oriented modeling language before we can use it.

Requirements for agent-oriented modeling tool

The first requirement is that BPMN must support the representation of an agent type. The concept of participants in the BPMN is similar to the concept of agent types in agent-oriented modeling. Hence, a BPMN pool that is used to represent a participant can be used to represent an agent type. The use of a BPMN pool has at least three advantages. First, the pool creates a clear visual boundary around an agent type which encapsulates its characteristics and behavior. A BPMN pool can be divided into a number of lanes (see Figure 3). This leads to the second advantage, i.e., it allows us to represent an agent type with multiple roles. Since the lanes within a pool can be organized into a hierarchical structure, we can represent the role hierarchy that an agent type may have, such as departments within a division and divisions within a firm. Finally, a BPMN pool implies the existence of domain control within the pool, which is in line with the notion of agent autonomy in agent-oriented modeling. Agent autonomy refers to the ability of an agent to make a decision independently (in contrast to having a central controller who directs all agents to do something).

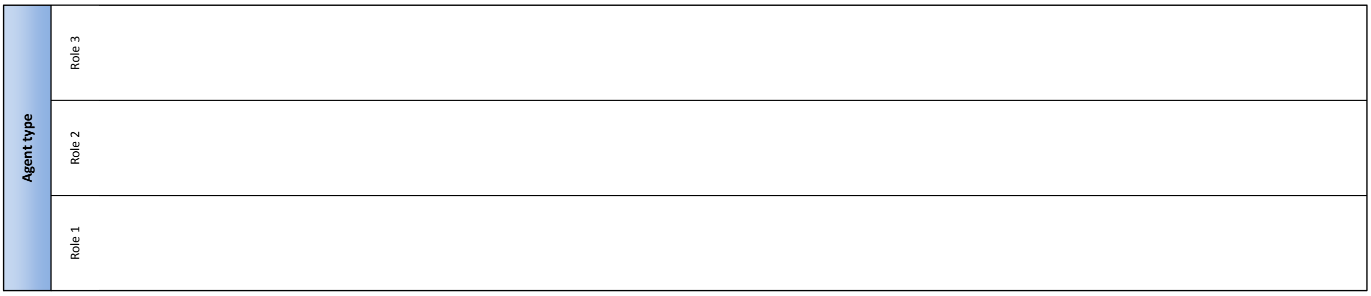

Figure 3: An agent type with multiple roles

As explained earlier, each agent type has a set of characteristics which can be static (such as birth date), relatively static (such as an organization's vision, mission and type of individual decision-making) or dynamic (such as memory and wealth). Hence, BPMN must have the capability to represent the characteristics of an agent type. This can be achieved using BPMN data annotation. BPMN data annotation allows us to show which activities use the characteristics and whether the characteristics are used as input or output. Figure 4 shows the characteristics of the patient which includes patient detail and payment detail. The details and agent type doctor are not shown to avoid cluttering this figure (in practice BPMN data annotation may be linked to a data dictionary).

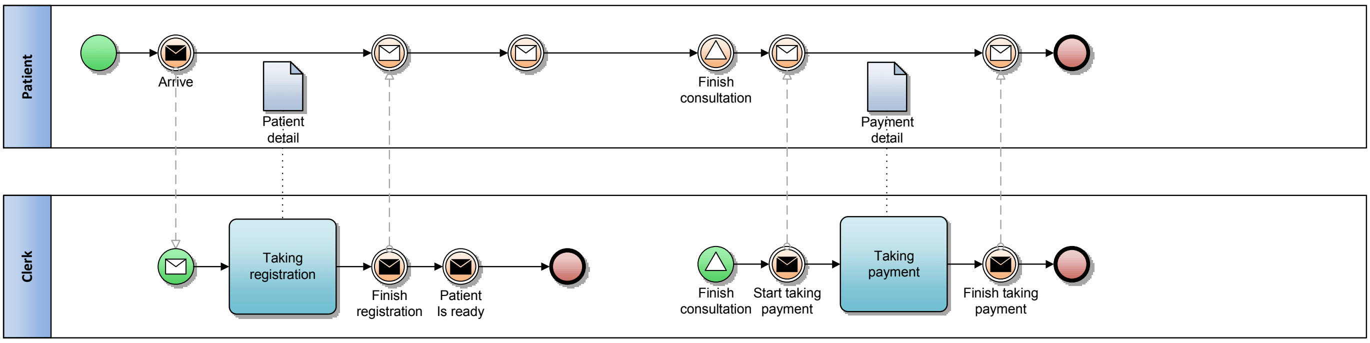

Figure 4: Characteristics of an agent type using BPMN

The next requirement is that BPMN must support the representation of agent's behavior. In BPMN, the behavior of an agent type can be represented using a combination of flow objects (events, activities and gateways) and connecting objects (the arrows/lines connecting the flow objects) encapsulated within the pool that represents the agent type. A BPMN event is represented as a circle in the diagram. It can be used to represent a start event (an event that triggers an agent to act), an immediate event (an event that is raised by an agent and triggers another action), or a final event that ends a certain process. In addition, BPMN allows us to specify whether an event will pause, resume or interrupt (and redirect) an action. A BPMN activity is represented using a rounded rectangle which is used to represent an action performed by an agent. It represents a step or action in the process. A BPMN gateway is represented using a diamond shape. It can be used to represent various types of decisions, such as branching, selection and merging. The BPMN connecting objects are used to connect two flow objects. Each connection denotes a sequence flow from one flow object to another flow object, a message passing flow from one agent type to another agent type, or an association between two BPMN objects. Experienced modelers may appreciate that it is possible to represent various complex agent behaviors using the combinations of BPMN flow objects and connecting objects. Figure 5 shows an example on how events, activities and gateways are used to represent a behavior (in this case, the detail of activity "taking payment").

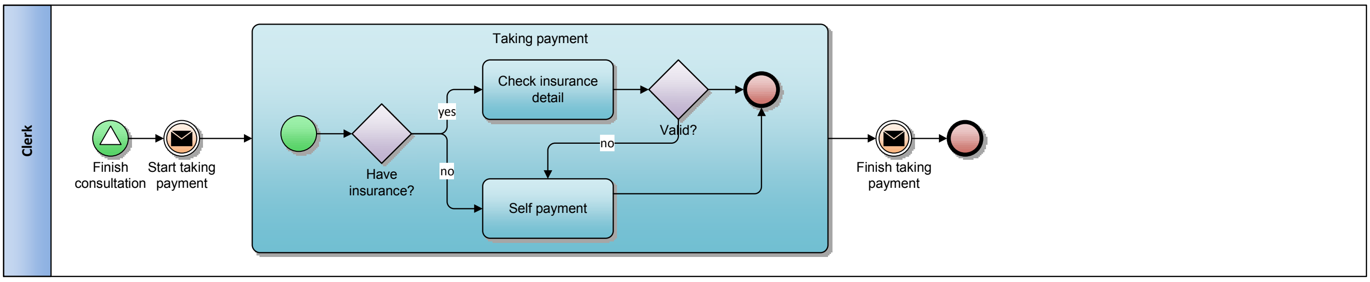

Figure 5: Example of the behaviour of an agent type using BPMN

Finally, all communications between participants in BPMN must be done using messages and signals. Messages are sent to specific participants and signals are broadcasted (and it is up to the participants to decide whether they are interested in those signals). This communication rule is consistent with how interactions between agents are done in agent-oriented modeling. Hence, we can show that BPMN has met the minimum requirements for an agent-oriented modeling tool.

Conclusion

In this white paper, I have shown that the same business process in an organization can be viewed using both process-oriented approach and agent-oriented approach. The main advantage of using the agent-oriented modeling approach is that we put humans including our customers and key assets/resources at the center of our model. Consequently, this approach should make us more aware of their objectives and concerns and incorporate them in our business process analysis and design.

This white paper has shown that we can use the current BPMN standard to represent an agent-oriented business process model (the business view of BPMN) by simply changing our view. We should note that the example used in this white paper is rather simple. The current BPMN standard may not be able to represent all types of agents and their interactions, especially if we would like to make the model executable (the technical view of BPMN). This is because the applications of agent-based modeling cover vast areas. Hence, it is difficult to prove that BPMN can be used for all agent-oriented models. However, the more we use BPMN to represent an agent-oriented model, the more we will learn on how BPMN can be used to represent the model more effectively. If we find a case in which BPMN cannot be used, we can then make a suggestion to improve the BPMN standard. This is how a standard evolves over time.

Bibliography

Object Management Group. (2010) Business Process Model and Notation (BPMN) version 2.0. Available from http://www.bpmn.org

Onggo, B.S.S. and Karpat, O. (2011) ‘Agent-based conceptual model representation using BPMN’, Proceedings of the 2011 Winter Simulation Conference, 11-14 December, Phoenix, Arizona. Los Alamitos, California: IEEE Computer Society Press. Available from www.informs-sim.org/wsc11papers/060.pdf

Onggo, B.S.S. (2012) ‘BPMN pattern for agent-based simulation model representation’, Proceedings of the 2012 Winter Simulation Conference, 9-12 December, Berlin, Germany. Los Alamitos, California: IEEE Computer Society Press. Available from http://informs-sim.org/wsc12papers/includes/files/con536.pdf

Onggo, B.S.S. (2013) ‘Agent-Based Simulation Model Representation using BPMN’, in Fonseca, P. et al. (Eds) Formal Languages for Computer Simulation: Transdisciplinary Models and Applications. Hershey, PA: Information Science Reference, pp. 378-399.

Orbus Software
3rd Floor
111 Buckingham Palace Road
London
SW1W 0SR
United Kingdom

+44 (0) 870 991 1851
enquiries@orbussoftware.com
www.orbussoftware.com

orbus
software