



Modeling Complex Business Environments for Context Aware Systems

P. M. Singh^(✉), L. P. Veelenturf, and T. van Woensel

OPAC Research Group, Department of IE&IS,
Eindhoven University of Technology,
Eindhoven, The Netherlands

{p.m.singh,l.p.veelenturf,t.v.woensel}@tue.nl

Abstract. Context awareness in complex business environments has been recognized as a major challenge for enterprise information systems. Although, the development of a context aware system in different application domains is convincingly documented in current literature, the design of such systems requires greater attention. Particularly, investigating the context of a context aware system. In this paper, we present a step-wise method to model a complex business environment. The method provides an approach for investigating a context and using the investigation results in subsequent design steps.

Keywords: Context aware · Business environment · System design · Complexity

1 Introduction

Advances in Information Technology (IT) have changed the nature of business environments. Today, business entities exist and work in an environment where they are interdependent and co-create value [1]. Some characteristics of such a business environment are, intricate value exchanges, a dynamic market, innovative products/services and loose customer loyalties.

By leveraging the advances in IT and computing capabilities, business entities adapt and customize their products/services to better align with customers' needs. In other words, they strive to provide *context-aware* products/services. Providing context aware services is no trivial task, particularly in today's business environments. Consequently, “*How to offer contextual capabilities in complex business environments?*”, is one of the four major challenges for Information Systems researchers highlighted by Kadiri et al. in [2]. In the same article, as a suggestion for future research, Kadiri et al. noted that “*Context models for complex enterprise applications.... are not being addressed in the research community so far*”. Similar observations have been made by Hong et al. in [3].

Contextual data is necessary to provide context aware products/services. However, the extraction of (relevant) contextual data in complex business environments require proper understanding of the environment [4]. In this paper, we

present a step-wise method to model complex business environments. Thus, our method will help system designers in the design of context aware systems.

In Sect. 2, Background, we clarify our understanding of a complex business environment. We also highlight the challenges presented by complex business environments for context aware services/products. Thereafter, in Sect. 3, we present our method which is divided in three phases, Motivation Modeling, Use Case Modeling and Situation Modeling. Section 4, illustrates our method using the case of Janssen Transport b.v, a transport company. The company desires dynamic routing of its trucks to provide context aware transportation services. Section 5 is the discussion section which provides a reflection over our proposed method. Finally, Sect. 6 concludes the paper and highlights the main takeaways.

2 Background

2.1 Context-Aware Systems

Context aware systems gather and analyze (relevant) contextual data, thereby aiding a business entity to provide context aware services [4]. Using such systems the business entity can adapt, modify, update or even change its products/services & underlying processes. Context aware systems are being developed and used in various application domains including; health-care [5], disaster management [6] and smart cities [7].

2.2 Complex Business Environments

According to Chen et al. [8] complexity in an industry, corresponds to the number, multiplicity and distribution of external factors. Thus, it refers to the heterogeneity and concentration of businesses within an industry. Based on their study of 746 business entities, they concluded that as the complexity of a business environment increases, it becomes increasingly difficult for managers to make strategic decisions. Saleh and Watson [9] in their study of VUCA¹ business environments, define complexity as, the interconnected parts, networks and procedures within the organization and within the external business environment. Both studies discussed above, approach business complexity from a rather, wide perspective. While Chen et al. discuss complexity of industries as a whole, e.g., the air conditioning (manufacturing) industry, steel industry; Saleh & Watson discuss complexity w.r.t long term strategic decision making.

Our understanding of business complexity, relevant for context aware services, is complexity at the *operational*, business process level. It is aligned to Vaconselos and Ramirez's [10] concept of *natural complexity*. According to them there are two types of complexities;

1. algorithmic complexity - the difficulty in solving a given, well-defined, problem.

¹ volatile, uncertain, complex and ambiguous.

- 2. natural complexity - characterized by no unique solution since the solution depends on the interpretation of the problem.

An example of algorithmic complexity is, to find all words in a dictionary which start with ‘q’ and end with ‘t’. An example of a natural complexity is, say, to find all words in a dictionary which a child can easily read. The correct answer to the first example is unique, a fixed list of words, irrespective of who searches for the words. However, there is no one correct answer for the second example. The list of words, in this case, depends on interpretations like, who is a *child*? *what* does *easily read* mean? For distinction between the two complexities, Vascocelos & Ramirez call algorithmic complexity as *complication* and natural complexity as *complexity*.

From the perspective of context aware services, a complex business environment is one where interpreting the system’s context, is a complex problem and cannot be achieved with certainty. Moreover, determining/computing the appropriate response based on contextual event(s) might involve high complication. The above discussion leads us to divide business environments into four broad categories as shown in Fig. 1. By proposing a method in this paper, our goal is to aid system designers in the interpretation of Category II and IV environments.

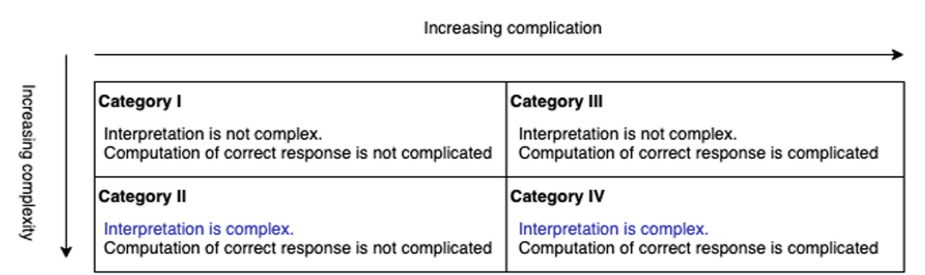


Fig. 1. Different business environments along with context interpretation and response computation in each of them. Text in blue indicates the focus of this paper. (Color figure online)

3 Method

This section presents our method for modeling complex business environments. Using our method, situation of interest for a context aware system can be interpreted and modeled. The method has three sequential phases, i.e., Motivation Modeling, Use-case Modeling and Situation Modeling. With each phase the level of detail about the system’s context increases, thereby enabling its interpretation. We have designed the method based on our experiences in different research projects aimed at designing context aware systems.

3.1 Phase I Motivation Modeling

Phase I, motivation modeling, provides a high level view of the current and desired scenario. Modeling them clarifies the motivation behind providing context aware services. The output of Phase I is a motivation model, linking the current scenario, to-be-provided services and the desired scenario.

Besides obvious value exchanges like money, products and services, business entities in a business environment also exchange intangible assets, e.g., knowledge, data, expertise and information. To correctly model all value exchanges in a complex business environment, a formal yet simple modeling technique is needed. Previous research, e.g. [1], shows that e-3 value modelling is an useful modeling tool for that purpose. It enables identification, modeling and analysis of all value exchanges between business entities in a complex business environment [1]. An e-3 value model consists of actors, value offerings and value transfers (Table 1).

Table 1. Text vs e-3 value model

Text	e-3 value model
Business entities	Actors
Users/customer	Actors
Products, services, knowledge, etc.	Value objects
Transactions between entities	Value transfers

- Activity 1: Model the current scenario of the complex business environment using an e-3 value model.
Rationale. To identify all relevant business entities in the environment and their value exchanges.
 - Activity 2: Model the desired scenario of the complex business environment using an e-3 value model. Add new *value objects* representing new services to be provided in the desired scenario.
Rationale. To identify (a) what is the additional value provided by context aware service, (b) who is the beneficiary (c) who (possibly more than one business entity) provides the new services and (d) which additional activities are needed to provide these services.
- A link between the two scenarios is needed to specify, which concerns in the current scenario are addressed by introduction of context aware services in the desired scenario. Although, such relationships can be (textually) deduced from the two e-3 models, however, their deduction may not be straight forward and explicit. Therefore, a modeling tool may be used which links the motivation (current scenario) and the goal (desired scenario). A motivation-goal modeling language, like ArchiMate 3.0 [11] is one such modeling tool.

- Activity 3: Make a motivation model linking the concerns in the current scenario to the goals of the desired scenario, using a motivation-goal modeling approach, e.g. Motivation Extension (ArchiMate 3.0) or Business Motivation Model (OMG) [12]

Rationale: Specifying the motivation to provide context aware services. By creating the motivation model, the intended value from context aware services in the desired scenario is better understood.

3.2 Phase II Use-Case Modeling

In Phase II, use-case modeling, results from Phase I are used to create a domain model and use case diagrams. A domain model includes both physical and abstract objects, thereby facilitating a clearer understanding of the context. It is derived from the motivation model and thus restricts domain modeling to relevant entities only. The use case diagrams build upon the domain model and the motivation model to further dissect (new) services. They provide the designer a concrete view of the interactions between the user, the system and the context.

- Activity 4: Create a domain model showing relevant entities, intrinsic and relational entities in the environment. An *Entity* is an object capable of independent existence. An *Intrinsic Entity* is a property of the entity and a *Relational Entity* is an entity which models the relationships between two entities.

Rationale: To document and model all relevant entities, their properties and relationships.

- Activity 5: Use the motivation model and domain model to create use case diagrams. Typical questions to be asked during this activity are (a) who uses the (new-context aware) services? (b) when are these services used? (c) do these services re-use existing services?

Rationale: A use case diagram clarifies the user-system interaction. It is a first attempt by the designer to understand the situations in which the system should provide its services.

3.3 Phase III Situation Modeling

In Phase III, situation modeling, results from Phase II are used to create situation models. Here, a situation refers to a *state of the environment* in which the user uses the context aware service.

- Activity 6: Using the use case diagrams write the situations in which the user, uses the services provided by the system.

Rationale: Documenting situations of interest.

- Activity 7: If Activity 6 produces new entities, add them to the domain model.

Rationale: A more comprehensive representation of the context. All entities in the domain model can together represent all situation in which the user uses the services.

- Activity 8: Based on the domain model, formalize the situations from Activity 6. Represent them using pseudo code or graphical situation models [13].
Rationale: To give a formal representation of the situation of interest in the business environment. Pseudo code or situation models are then converted to executable code.

At the end of Phase III, the designer has modeled situations of interest in the business environment. They are represented in terms of entities and their properties. It is these situations which a context aware systems should detect and interpret. By employing the above steps the system designer has a methodical approach for modeling a complex business environment. The following section illustrates the method using the case of Janssen Transport b.v., a transport company desiring dynamic routing of its trucks.

4 Illustrative Case Study - Janssen Transport b.v.

4.1 Current Scenario

Janssen Transport b.v. (JT) is a transport company in The Netherlands. It owns trucks and provides transportation services to retail stores and laundry businesses. Frank Cleaners (FC), a laundry business, is a client of JT. FC washes laundry from hotels, restaurants, vacation parks and dry cleaning stores. The collection of dirty laundry is outsourced by FC to JT. All Hotels² to be visited by JT, have been divided into five *sectors* by JT's management based on their proximity to each other. Every day one truck is assigned to a sector. The truck driver collects dirty laundry from all hotels in that sector and brings it to FC's Cleaning Unit. The sequence in which a truck visits the hotels in a sector, i.e., a sector's route, is fixed and is decided by JT's management. Hotels farthest to the cleaning unit are visited first.

The quantity of dirty laundry to be picked at a hotel, is not known in advance and varies considerably per day. Frequently, during their routes, trucks become full, while some hotels still need to be visited by them. The trucks then go to the FC cleaning unit, off-load the dirty laundry and continue further to visit the remaining hotels. Traffic conditions affect the arrival time of trucks at hotels. If a truck is delayed en-route to a hotel, often it is delayed to all subsequent hotels on the route, i.e., a *domino effect*. Hotels have a fixed time window (08:00 h–14:00 h), during which they allow their laundry to be picked up by the trucks.

4.2 Desired Scenario

JT has the following concerns over its client FC.

1. Numerous instances of unpicked laundry from hotels, a direct monetary loss for JT.

² further on we will use the term Hotels to refer to all collection points for laundry.

2. Late arrival of trucks at the hotels, i.e., beyond the time window. Delays are primarily due to traffic conditions. Due to delays, the same hotels, those visited last in each route, get affected. In the desired scenario JT wants to adopt a pro-active approach of routing its trucks based on historical travel time data and real time traffic data.
3. While some trucks, at the end of their route, return to FC Cleaning Unit half empty (unused free capacity); others offload laundry and have to resume on their routes. In the desired scenario JT wants to predict the daily demands of each hotel and optimize the truck routes. Demand prediction will be based on historic demand data.

To address above concerns, JT requires a context aware routing system. The system should dynamically route trucks, optimize truck capacities and minimize delayed arrival of trucks.

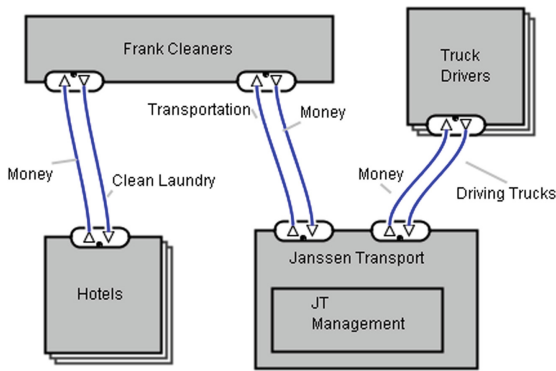


Fig. 2. Janssen transport current scenario

4.3 Phase I: Motivation Modeling

- Activity 1: Fig. 2 shows the e-3 value model for JT's current scenario. It models the business entities in the environment and the services they offer.
- Activity 2: Fig. 3 shows the e-3 value model for JT's desired scenario. In it, JT provides better transportation services to FC. Instances of unpicked dirty laundry decrease and trucks arrive at hotels on time. These are achieved via the new value activity, Context aware dynamic routing. Beside driving, truck drivers now provide their location and amount of laundry picked up at each hotel to JT. Data providers furnish real time traffic information to JT used for ETA calculation of its trucks.
- Activity 3: Fig. 4 show the Motivation Model based on Activity 1, 2 and textual descriptions. In the model (a) Assessment elements model problem-s/concerns of JT based on the current scenario. (b) Driver elements model

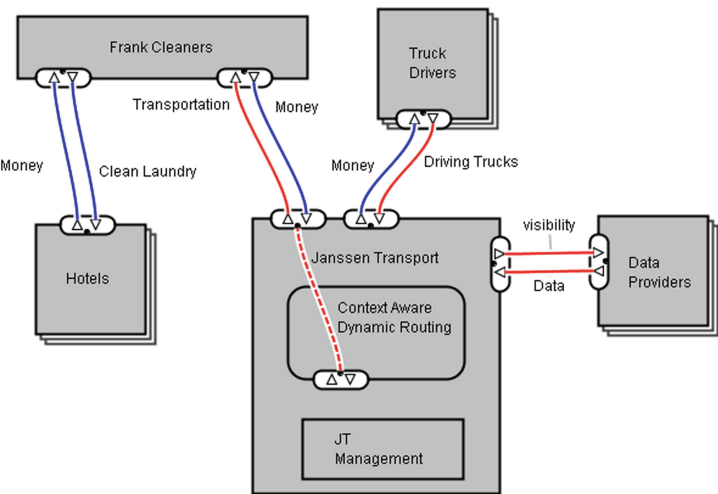


Fig. 3. Janssen transport desired scenario

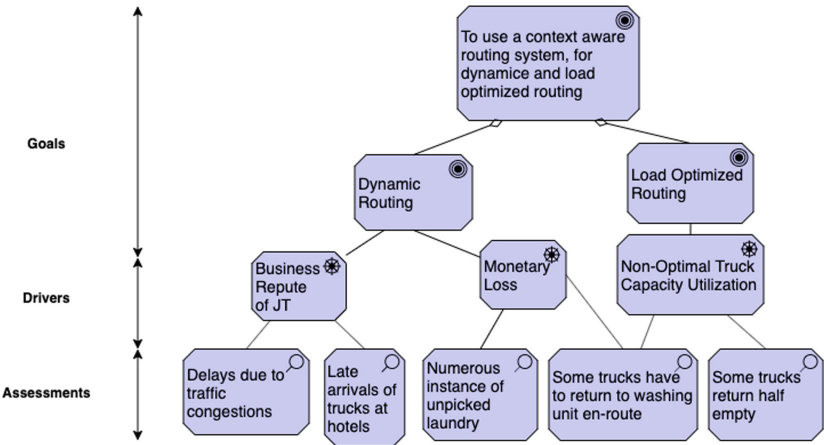


Fig. 4. Motivation model using ArchiMate 3.0 motivation elements (assessment, drivers and goals)

the relevance and effect of each problem/concern. (c) Goal elements model the new/improved service provided in the desired scenario. Each goal caters to at least one driver or a lower level goal.

4.4 Phase II. Use-Case Modeling

- Activity 4: The motivation model enables us to design a simple domain model (Fig. 5). Entities are in **Bold**, intrinsic entities in blue and relational entities in red.

- Activity 5: The motivation model and domain model enable us to make a simple use-case diagram of the Context Aware Routing System (Fig. 6). To determine a truck's delay, the truck's location is provided by the truck driver after logging in the system. The location is used for dynamically routing the truck (Goal). When the truck arrives at a hotel, the driver enters the arrival time and the picked up laundry in the system. The total capacity of the truck and cumulative load of laundry picked up is required for Load Optimized Routing (Goal) of the trucks. The use-case diagram thus connects the system's services to goals in the desired scenario.

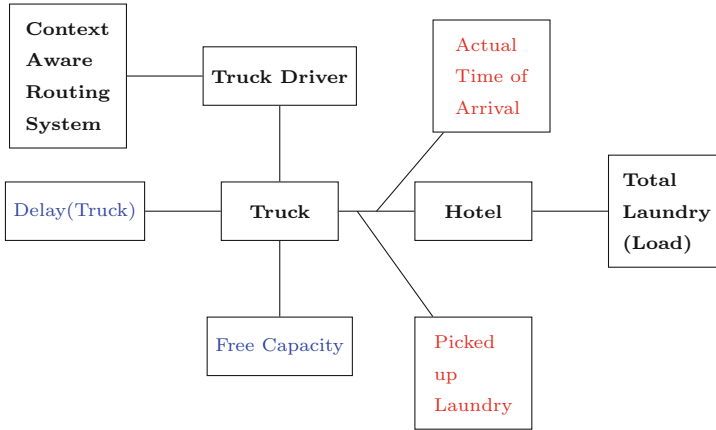


Fig. 5. Domain model (Color figure online)

4.5 Phase III. Situation Modeling

- Activity 6: The following three situation in the business environment have to be detected by the context aware routing system.
 - **Situation 1.** A truck's route in a sector is not the fastest route and remaining hotels can be visited earlier via a different route. To determine this situation, historical travel time data and real time traffic data is required.
 - **Situation 2.** A truck might have to return to the cleaning unit during its route. In other words, the current free capacity of the truck is less than the predicted load of the remaining hotels. A different truck should pick up the load from such hotels. In Fig. 7, (top) Truck 1 becomes full after loading laundry from Hotel C, and still has to visit Hotel D. The subsequent route for Truck 1 in the current situation is shown highlighted using red arrow. In the desired scenario (bottom) when Truck 1 becomes full, Truck 2 from an adjacent sector visits Hotel D.

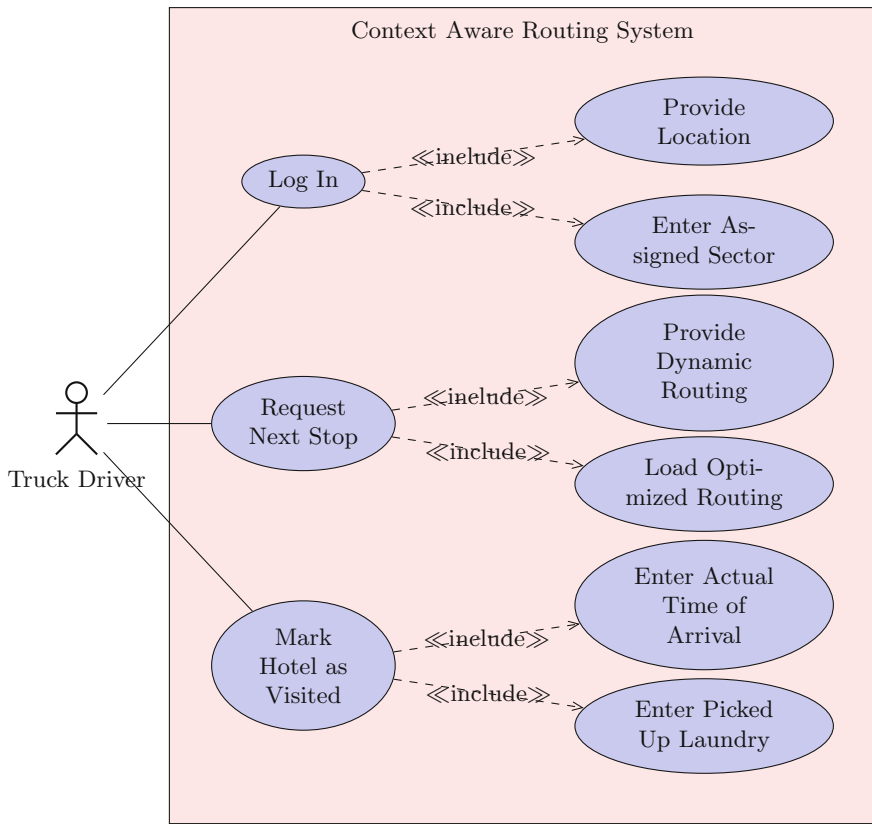


Fig. 6. Use case diagram

- **Situation 3.** A truck can not arrive at hotel(s) within the time window. It should be possible for another truck to pick-up laundry from such hotel(s). In Fig. 8 (top) Truck 2 is en-route between Hotel E and Hotel F. The road has traffic congestion, thus traffic movement is slow. Though Truck 2 can reach Hotel F on time it will not be able to do so for Hotel G. In the current scenario, dirty laundry from Hotel G would be left uncollected. In the desired scenario another truck picks up dirty laundry from Hotel G, as shown in Fig. 8 (bottom).
- Activity 7: After describing Situations 1–3 in textual form, new entities were discovered and have been added in the domain model (Fig. 9).
- Activity 8: The three situations from Activity 6 are formalized (pseudo code) using entities from the updated domain model.
- **Situation 1.** At time t , elements of the set $RemainingHotels_{Truck}(t)\{\}$ are all hotels still to be visited by a truck. It is equal to all hotels in the sector assigned to the truck minus $VisitedHotels$. Thus, when a truck, say Truck B, leaves JT in the morning, $RemainingHotels_{TruckB}\{\}$

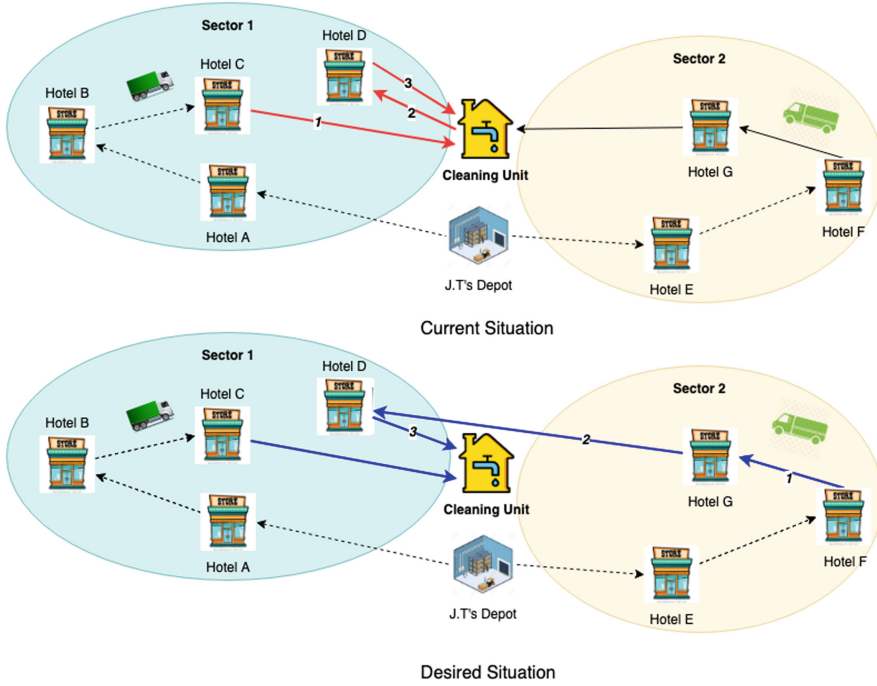


Fig. 7. Situation 2, when a truck is full en-route a different truck visits the remaining hotels

consists all hotels in the assigned sector. The route of a truck, at any given time, is represented by the sequence of hotels in the set $RemainingHotels_{Truck}(t)\{\}$. When the set $RemainingHotels_{Truck}(t)\{\}$ is empty, the truck goes to Cleaning Unit.

Given a route, at t , $Route_{Truck}(t)$; a faster route, $Route_{FasterTruck}(t)$, exist, if the ETA for $RemainingHotels_{Truck}$ via $Route_{FasterTruck}(t)$ is earlier than via $Route_{Truck}(t)$. Situation 1 is thus formulated as:

$$\exists Route_{Truck} \text{ such that } ETA(Route_{Truck}) < ETA(Route_{Current})$$

At any time t , there may be multiple routes such that the ETA for hotels is earlier than the current route. We may call the set of all faster routes at any given time t as $FasterRoutes_{Truck}(t)\{\}$. The system selects the fastest route, $Route_{Fastest}$, from the set $FasterRoutes_{Trucks}(t)\{\}$.

- **Situation 2.** In this situation, $FreeCapacity_{Truck}(t)$ is less than the total $PredictedLoad$ for $RemainingHotels_{Truck}(t)\{\}$.

Figure 7, shows a special case of Situation 2, where the $FreeCapacity = 0$ and Hotel D still needs to be visited, i.e., $RemainingHotels_{Truck}(t) = \{HotelD\}$. At any time (t) , Situation 2 can be formulated as follows:

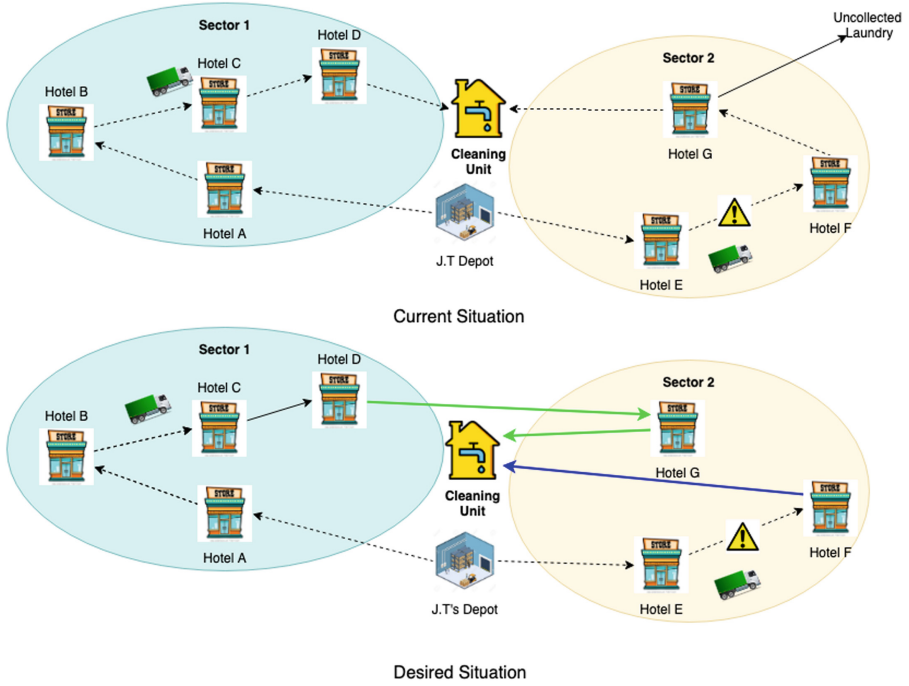


Fig. 8. Situation 3, when a truck cannot arrive at a hotel on time, a different truck visits the remaining hotels

$$FreeCapacity_{Truck}(t) < PLoad(RemainingHotels_{Truck}(t), day) \wedge \\ RemainingHotels_{Truck}(t) \neq empty \\ where$$

$$RemainingHotel_{Truck}(t) = \{HotelM, HotelN,\} \\ and \\ PLoad(RemainingHotels_{Truck}(t), day) = PLoad(HotelM, day) + \\ PLoad(HotelN, day) + \\ PLoad(HotelP, day) + \\ \quad (1)$$

If the above situation is detected, the system assigns truck(s) to collect laundry from the remaining hotels of the truck with limited capacity. The assignment of trucks is non-trivial and will depend on various conditions, including; cost, distance of the truck and its free capacity etc.

- **Situation 3.** In situation 3, the ETA of some trucks at remaining hotels is beyond the time window. In this situation, to optimize capacity utilization, the routing system assigns trucks to collect laundry from those

hotels. At time t , Situation 3 can thus be formulated as follows:

$$ETA_{Truck}((Location(Truck(t)), Hotel_i)) \text{ beyond } TimeWindow \quad \exists Hotel_i \in RemainingHotels_{Truck}\{\} :: \quad (2)$$

For calculation of ETA, traffic events and associated delay related to each event is needed.

After Activity 8, we have modeled situations of interest in terms of contextual data from JT environment. The system designer can use these in the subsequent design steps. The contextual data is in turn, itself derived from the domain, use-case and motivation models from previous design steps.

5 Discussion

JT's business environment is a *more complex* and *more complicated* business environment (Category IV, Fig. 1). The identification of situations, is based on historic data (e.g., travel times, pick up load) and real time data (e.g., traffic, truck capacity). However, such an identification cannot be done with certainty as travel times and pick up load are stochastic, not deterministic. The context aware routing system would constantly monitor the context for detecting the situations. Furthermore, the situations are difficult to identify via manual monitoring of the context, thereby making it a complex environment. The following points make the desired scenario *more complicated*, i.e., the computation of the correct system response, difficult:

1. Computation of the fastest route for each truck, using historic travel time data and real time traffic information.
2. Calculation of the monetary loss owing to unpicked dirty laundry from a hotel.
3. The ETA of a truck at a hotel can not be determined with certainty.
4. Selection of a truck among all trucks, such that extra costs associated with visiting additional hotels (e.g. fuel, driver compensation, etc.) is less than the loss incurred by unpicked laundry.

In Sect. 1, Introduction, we used the article by Kadiri et al. [2] to motivate our research. In the same article the authors had presented a generic model showcasing four layers of a context processing life cycle, i.e., Acquisition, Modeling, Processing and Dissemination. Table 2 compares these layers vis-à-vis the phases in our method. Phase III maps to the Processing Layer partially, since only Activity 8 delivers *codes* and *situation models* suitable for processing by development tools. Our method doesn't prescribe how situations of interest be communicated to the actors/business entities in the context. Consequently, there is no apparent mapping between the Dissemination layer and a phase in our method. Further study is needed to assess the above comparison and it is an area for future research. Additionally, future researchers should investigate the domain specific modifications needed for the method.

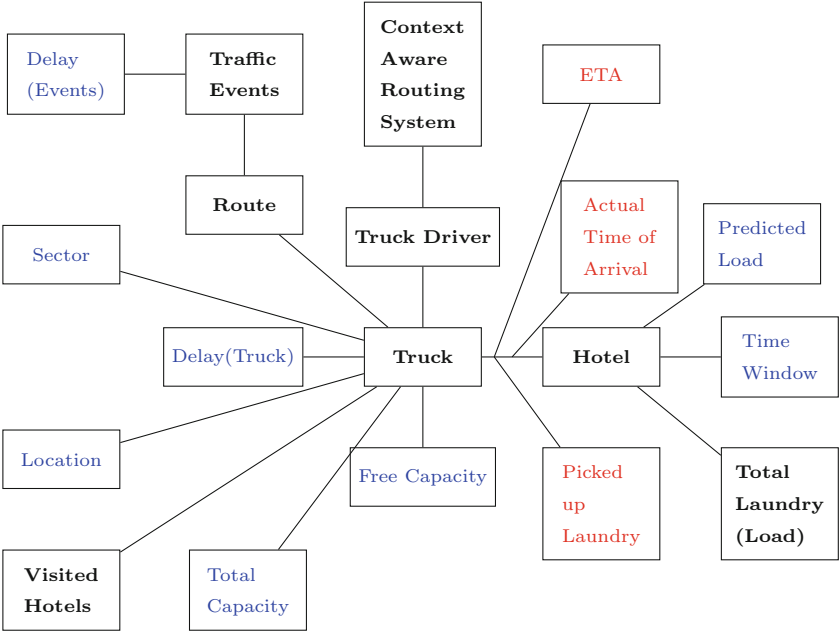


Fig. 9. Updated domain model

Table 2. Context processing lifecycle in [2] vs. Phases of the method

Layers [2]	Description in [2]	Phases in the method
Acquisition	Data and process contributing to context instance definition	Phase I
Modeling	Context representation in structured form	Phase II & III
Processing	Fusion of data and modeling with processing and reasoning mechanism	Phase III (partially)
Dissemination	Distributing the processed high-level context to context-consuming actors	No apparent mapping

Our method will enable system designers to avoid two pitfalls during context aware system design; (a) considering significant amount of contextual information during the initial design steps which later proves to be less relevant and (b) not including relevant contextual data in system design. Using our method, the designer is not overwhelmed with the complexity of business environments, thus avoiding the above pitfalls.

The proposed method is based on our experiences in system design and development. We did not follow a specific design methodology for designing

the method. This a limitation of our research as design choices made by us may not be explicit.

6 Conclusion

Complex business environments pose unique challenges for the design of context aware systems. In this paper we presented a step-wise method to model a complex business environment. Via our method we, (a) provide a direction for investigating the context and (b) highlight the use of investigation results in subsequent design steps. System designers will find our method useful for the design of context aware systems in complex business environments. Future researchers should apply the method in varied application domains which would lead to further improvement of the method.

References

1. Wieringa, R., Engelsman, W., Gordijn, J., Ionita, D.: A business ecosystem architecture modeling framework. Paper presented at the 21st IEEE Conference on Business Informatics (CBI) 2019, Moscow, Russia (2019)
2. Kadiri, S.E., et al.: Current trends on ICT technologies for enterprise information systems. *Comput. Ind.* **79**, 14–33 (2016)
3. Hong, J., Suh, E., Kim, S.-J.: Context-aware systems: a literature review and classification. *Expert Syst. Appl.* **36**(4), 8509–8522 (2009)
4. van Engelenburg, S., Janssen, M., Klievink, B.: Designing context aware systems: a method for understanding and analysing context in practice. *J. Logical Algebraic Methods Program.* **103**, 79–104 (2019)
5. Trinugroho, Y.P.D., Reichert, F., Fensli, R.: An ontology enhanced SOA-based home integration platform for the well being of inhabitants. Paper presented at the 4th IADIS International Conference on e-Health 2012, Lisbon, Portugal (2012)
6. Fleischer, J., et al.: An integration platform for heterogeneous sensor systems in GITEWS - Tsunami Service Bus. *Nat. Hazards Earth Syst. Sci.* **10**, 1239–1252 (2010)
7. Auger, A., Exposito, E., Lochin, E.: iQAS: an integration platform for QoI assessment as a service for smart cities. Paper presented at 3rd World Forum on IoT 2016, Reston, USA (2016)
8. Chen, H., Zeng, S., Lin, H., Ma, H.: Munificence, dynamism, and complexity: how industry context drives corporate sustainability. *Bus. Strategy Environ.* **25**, 125–141 (2017)
9. Saleh, A., Watson, R.: Business excellence in a volatile, uncertain, complex and ambiguous environment (BEVUCA). *TQM J.* **29**(5), 705–724 (2017)
10. Vasconcelos, F.C., Ramirez, R.: Complexity in business environments. *J. Bus. Res.* **64**, 236–241 (2011)
11. ArchiMate 3.0.1 Specification. <https://pubs.opengroup.org/architecture/archimate3-doc/chap06.html>
12. OMG Business Motivation Model v1.3. <https://www.omg.org/spec/BMM/1.3/PDF>
13. Costa, P.D., Mielke, I.T., Pereira, I., Almeida, J.P.A.: A model-driven approach to situations: situation modeling and rule-based situation detection. Paper presented at the IEEE 16th International EDOC Conference 2012, Beijing, China (2012)