Towards Context-aware Border Security Control

Boris Shishkov[1,2] and Dimitris Mitrakos[3]
[1]Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria
[2]IICREST, 53 Ivan Susanin Street, 1618 Sofia, Bulgaria
[3]Department of Electrical Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
b.b.shishkov@iicrest.org, mitrakos@eng.auth.gr

Keywords: Enterprise Modeling, Model-driven Engineering, Context-awareness, Border Security, SDBC.

Abstract: Context-aware systems allow for adapting the system behaviour to the context situation at hand and we have seen good applicability of context-aware systems in domains, such as Mobile Health. Even though this could also be useful for the Border Security domain, applying context-awareness in this domain is not trivial since the possible context situations are numerous and difficult to predict. Still, context-aware Border Security systems are needed as a possible way to overcome the inevitable shortage of resources along the borders. Smooth and fast border crossing for travellers, in combination with adequate level of security, can be achieved if: (i) at any moment, the context situation is properly captured; (ii) there is potential for behaviour and resources (from the Authorities' side) corresponding to each possible context situation. The context situation capturing is about sensors, data streaming, and so on. Establishing the right behaviour / resources is about enterprise modeling and business rules, and it is also about automation that assumes in turn integration of software applications in the overall Border Security system. In this position paper, we address all that, inspired by the SDBC Approach, Enterprise Ontology, Semiotic Norms, and the principles of Context-aware Systems. Reporting research-in-progress, we only present our way of modeling and we identify several domain-specific concerns that are related to the application of the built models. We also provide a list of recommendations that are expected to be useful with regard to possible Border Security system developments.

1 INTRODUCTION

Most software systems need to be adequately integrated in their enterprise environment comprising enterprise modules, human agents, and even already running software applications. Hence, traceability and possible mappings are often needed between a piece of software and an enterprise module and vice versa (Shishkov, 2016); this we call *enterprise-software alignment* (Shishkov and Janssen, 2016). The SDBC Approach allows for realizing such an alignment in a component-based way (Shishkov, 2005), justifying that software specification should be based on corresponding enterprise models. This can be achieved when we have sufficient knowledge of the enterprise environment and this environment is more or less stable. Nevertheless, if we have a constantly changing enterprise environment, we would face a new challenge – the challenge of handling *adaptability*. This means adapting to the surrounding situation in two directions: firstly, if the enterprise environment is changing, then the enterprise modules "inside" should be changing in turn; secondly, if the enterprise environment is changing, then the "behaviour" of the software applications running inside the enterprise should be adapted as well. Said otherwise, we should be able to adequately establish the situation at hand and provide the right enterprise / software behaviour accordingly. This was called *context-awareness* in research addressing such adaptability-related challenges in the domain of Mobile Health (AWARENESS, 2006). An example is Tele-Monitoring: a patient is being monitored from distance, using mobile technology; there are several possible-to-occur situations, each of them being easy-to-capture distantly; then action is triggered, corresponding to the captured context situation. Thus, we argue that context-awareness is a solution and is easy-to-apply only if we have a limited number of possible-to-occur situations. However, in the current position paper, we address a domain characterized by MANY possible-to-occur situations: this is the `Border Security` domain. In particular, we are interested in the *monitoring of illegal migration and in combatting related crime*

207

Shishkov B. and Mitrakos D.
Towards Context-aware Border Security Control.
DOI: 10.5220/0006223802070212
In *Proceedings of the Sixth International Symposium on Business Modeling and Software Design (BMSD 2016)*, pages 207-212
ISBN: 978-989-758-190-8

(and goods smuggling). Guarding along country borders by electronic means, using various channels: infrared images, visible images, proximity sensors, and so on, followed by some kind of intelligent data fusion algorithms, has been addressed in several European Projects, such as the European FP7 Project: "Protection of European Seas and Borders through the Intelligent Use of Surveillance" (PERSEUS, 2015) and the European EBF Project: "Land Border Surveillance – Strengthening of Reaction Capacity" (LandBorderSurveillance, 2012). Still, those efforts appear to be insufficient as it can be seen from the severe border problems in Greece and Italy, in 2015-2016 (FRONTEX, 2016). We recognize the need for better interoperability with regard to the existing national border security platforms and systems. We do agree that innovative capabilities are needed, including trans-national exchange of available and useful information, as it has been discussed in the above-mentioned projects. Nevertheless, we argue that those would only be part of the solution related to the challenge of improving Border Security, supported by (software) technology. We claim that context-awareness is not applied of full value at the borders and we claim that context-aware Border Security systems would make a difference in the direction of improvement. At the same time, we go back to our previous conclusion, already mentioned, that it is not straightforward applying context-aware solutions in the Border Security domain. Hence, research is needed on *Context-Aware Border Security* (CABS) control, as a possible way to overcome the inevitable shortage of resources along the borders (it would be difficult for a country to supply persons and equipment at every potentially risky border point). A CABS system would guarantee adaptability with regard to the situation at hand – persons and equipment would only be supplied at the spot where they are needed and in the moment when they are needed. Of course, if total "tension" appears at all risky border points at the same time, such an approach would "crash" but this is similar to the situation of all customers of a bank claiming back their deposits at the same time. Such situations are considered to be of low probability to occur and are thus left beyond the scope of this paper.

Hence, we claim that smooth and fast border crossing for travellers in combination with adequate level of security can only be achieved if:

(i) at any moment, the context situation is properly captured;
(ii) there is potential for behaviour and resources (from the Authorities' side) corresponding to each possible context situation.

The context situation capturing is about sensors, data streaming, and so on. Establishing the right behaviour / resources is about enterprise modeling and business rules, and it is also about automation that assumes in turn integration of software applications in the overall Border Security system.

Further, because of the increased complexity with regard to Border Security situations and occurrence probabilities, we need to address *data aspects* (going beyond just interoperability and information exchange, see above). Ways to capture data, quality of data and the probability that the captured data is correct, reliability, versioning, privacy, and so on, are of importance as well. Hence *data analytics is to be integrated in the enterprise modeling and also in the software development* in order to facilitate context-awareness, especially in the Border Security domain.

In this position paper, we address all this, inspired by the SDBC Approach (Shishkov, 2005), Enterprise Ontology (Dietz, 2006), Semiotic Norms (Liu, 2000; Shishkov et al., 2006), and the principles of Context-aware Systems (AWARENESS, 2006).

Reporting research-in-progress, we only present our way of modeling (Section 2) and we identify several domain-specific concerns that are related to the application of the built models (Section 3). We also provide (as part of the conclusions) a list of recommendations that are expected to be useful with regard to possible Border Security system developments.

2 WAY OF MODELING

As mentioned above, we address the challenges of deriving software and integrating it in its enterprise environment, inspired by the SDBC Approach (Shishkov, 2005). This in turn assumes reference to the theories of *LAP – Enterprise Ontology* (Dietz, 2006) and *Organizational Semiotics* (Liu, 2000); those are briefly outlined in another paper published in the current proceedings (Shishkov & Janssen, 2016). The idea behind SDBC is that (re-usable) enterprise modeling constructs (called "Business CoMponents") are identified and reflected in corresponding software specification models to be in turn decomposed in terms of models of software components. SDBC is consistent with the principles of Model-Driven Engineering – MDE (Schmidt, 2006): building a technology-independent model goes first, then it is to be decided what would be automated, and in the end is the software derivation. This is the "basis", no matter if we go for developing a context-aware system or a system that is not

Towards Context-aware Border Security Control

context-aware – in order to contribute especially in the direction of context-awareness and particularly in the Border Security domain, we need to have a "valid" basis to start to build upon. For that we take SDBC not only because developing this approach is part of our previous work but also because SDBC has been validated by means of case studies carried out in the domains of Finance and Healthcare.

Further, the SDBC Approach and the principles of MDE were successfully applied in specifying context-aware systems in domains, such as Mobile Health.

Nevertheless, as mentioned already, the Border Security domain assumes greater complexity because of numerous possible situations and prediction difficulties. Further, what is observed at the border is a "mixture" of personnel and devices, subject to numerous rules and functionalities (FRONTEX, 2016), and it is not trivial approaching this in terms of technology-independent models, automation, and so on. This is because some (intuitive) tasks can only be realized by humans while other (surveillance) tasks can only be realized by devices, to give just an example. Hence, we need to "adapt" SDBC to the peculiarities of the Border Security domain. SDBC goes "top-down", from a "bird-view" enterprise model through delimitation with regard to the software system –to-be, to implementation. Nevertheless, for specifying a CABS system, we propose to go "middle-out", as exhibited on Figure 1, and we adapt the application of SDBC accordingly.

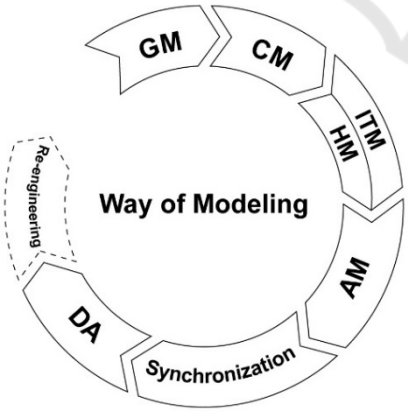

Figure 1: CABS – Way of Modeling.

On the Figure: "GM" stands for "*general model*", "CM" stands for "*conceptual model*", "ITM" stands for "*IT model*", "HM" stands for "*humans model*", "AM" stands for "*aspect model*", and "DA" stands for "*data analytics*".

We propose to go "middle-out" because in the Border Security domain, it seems most pragmatic to start with modeling "what is there" (a mixture of person-tasks, device functionalities, and so on to be seen at the border) – such a model we call a *general model* (GM). No other model that would inevitably be abstract, would allow for grasping everything correctly and also communicating it adequately with all relevant stakeholders – this is claimed to be of great importance particularly for the Border Security domain. Just as an example of GM, we consider a typical point at an external EU border, the border between Bulgaria and Turkey (FRONTEX, 2016), and we take an "imaginary" view on things that may be seen at a border point – see Figure 2.

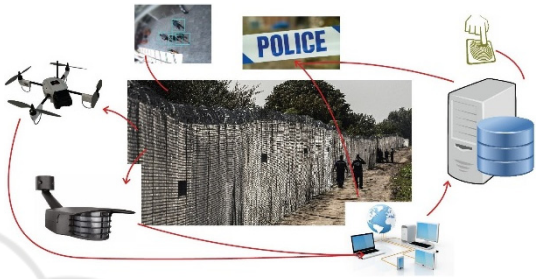

Figure 2: GM Example.

As seen from the Figure: there is a border fence and border police officers patrolling along the fence; there are cameras attached to the fence, which realize crowd monitoring and there are mobile cameras attached to drones; there are finger-print devices that can be used by police officers for personal identification; there are (networked) computers running and streaming all sensor raw data, and also processing it by applying data fusion algorithms (for example), allowing "higher-level" reasoning, and so on. Hence, we claim that such a model should be the starting point in specifying a CABS system.

We use the GM as basis for deriving a CABS-related classification of concepts – this we call a *conceptual model* (CM) – see Figure 1. This way of "arriving" at the CM guarantees that our further system development activities would be "grounded". The `human agent` concept and the `device` concept appear to be essential within the CABS conceptual model (Figure 3). That is because the CABS general model suggests that anything that can be observed at the border either relates to a personal (human) role or to a functionality delivered by a device (equipment). Further, among the human agents at the border (besides the persons who are crossing the border and are thus left outside the scope of the CABS system) are `customs officers` and `police officers`, while among the devices one could observe at the border are `sensors`,

209

computers, and vehicles. Sensors in turn could be audio sensors and video sensors, while computers could be servers and personal computers, and vehicles could be cars and drones. And so on. This is just as an example on how a CM can be derived, based on a GM.

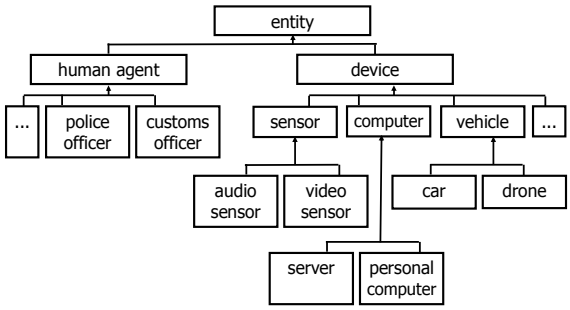

Figure 3: Deriving a Conceptual Model.

Such a conceptual model is the necessary starting point in an SDBC software development but also if one would just need to build an enterprise model.

What goes next, as it can be seen from Figure 1, is splitting in two, namely: *humans model* and *IT model*. Said otherwise, all that concerns persons at the border is put on one side and all that concerns devices at the border is put on another side. We do not mix this up even though in enterprise modeling we often mix up activities that are realized by humans and activities that are automated, realizing that it is possible to further automate activities currently realized by humans. For example, the SDBC Approach was used to specify the automation of what a human actor (in particular: Insurance Broker) was doing, for the sake of "replacing" humans by software (Shishkov, 2005). In such cases, it is straightforward putting together in the model issues that concern human actors and issues that concern non-human actors. This is claimed to be nevertheless inappropriate with regard to the Border Security domain and that is why the CABS conceptual model strongly distinguishes between human agents and devices (see Figure 3). Hence, as according to Fig. 1, we model the human entities and processes (HM), and in parallel, we model the devices-related entities and processes (ITM).

That is because devices at the border (on one hand) capture raw data, stream data, run algorithms, and so on, and all those issues concern the electronics features of the devices, not even so much the software in some cases; further, those societally elementary but computation-intensive tasks assume nothing like real-life communication, intuitive behaviour, pro-activity, and so on. Persons at the border (on the other hand), such as border police officers, for example, are often valued especially for their intuitive behaviour, enriched by many years of experience. It is the person (not the device) who can "smell in the air" anything wrong and possibly trigger a check without even being able to explain why. This is in contrast to other domains where sometimes a task can equally be completed by a human actor or by a non-human actor. We claim that at the border, what persons do and what devices do are "two different Worlds".

What is next (see Figure 1) is the modeling elaboration – each of the two models (HM and ITM) is to be elaborated; having provided just a classification of concepts is not enough - statics (entities and their relationships), dynamics (processes and states), and so on, need also to be provided as elaboration. Such elaboration models are called *aspect models* (AM). We build those models, inspired by the four modeling perspectives of the SDBC Approach: *Structural Perspective* that reflects entities and their relationships (Dietz, 2006); *Dynamic Perspective* that reflects the overall business process and the states of each entity (evolving accordingly) (Van der Aalst, 2011); *Data Perspective* that reflects the information flows across entities within the system and flows reaching beyond the system's boundary (Shishkov, 2005); *Language-Action Perspective* that reflects real-life human communication and expression of promises, commitments, and so on (Shishkov et al., 2006).

Hence, as elaboration, we build structural models, dynamic models, data models, and language-action models – those are four aspect models. We proceed as follows:

- With regard to the HM we do all four elaborations, but
- With regard to the ITM, we do all except the last one – the language-action model, because devices cannot express commitments, do promises, and so on, using language.

Then, after having elaborated the HM and the ITM in terms of aspect models, we need to "synchronize" between the two, as depicted in Figure 1. By bringing together the two structural models, we establish all relationships between human entities and device entities, by bringing together the two dynamic models, we establish if any human action requires as pre-condition a corresponding device action to be completed (and vice versa), and so on. A question could be asked why was it necessary to firstly split and then synchronize. By firstly splitting in two (persons and devices) and then bringing them together, we guarantee for:

Towards Context-aware Border Security Control

- Adequate modeling, because each model is being generated based on the right background (when modeling humans, we consider humans and when modeling devices, we consider devices);
- Exhaustive modeling, because in the end the "two parts" are brought together.

Then comes the *data analytics* (DA) – see Fig. 1 and it is important not to confuse this with the "data perspective" of SDBC, reflected in deriving the aspect models. This perspective has a purely functional drive – we consider the data flows as part of the delivered functionality (for example, a balance needs to be checked, before allowing a person to withdraw money from his/her account). The DA, on the contrary, has a non-functional drive and is crosscutting. For example: (i) Extracting a face image of a person, allows for realizing activities on that basis but is it legally correct to do so and if not, is this information an "input" or not? (ii) If what we get as sensor raw data has, say 80% trustworthyness, what we do forward? Those are just two examples and we would also mention data versioning, obsolete data, wrong approximations, and so on. Nevertheless, just staying aware of those issues is not helpful – we need to address them functionally (even though in essence they may be non-functional). A possible way to do this is by applying Semiotic Norms – see another paper in the current proceedings (Shishkov & Janssen, 2016), for example (on considering crowd monitoring):

`Whenever` the System has established a situation
`if` the level of trust is less than 75%
`then` the System
`is` obliged `to` check also another source.

And in the end comes *re-engineering* – Figure 1, and the line there is dashed, to indicate that it is not always needed to re-engineer the system. Still, if the built models help in identifying inconsistencies, re-engineering could follow straightforwardly.

This is our proposed way of modeling CABS systems, in line with the SDBC Approach and as already mentioned, in the next section, we reason about the application of the CABS models, by identifying several concerns related to such an application.

3 DOMAIN-SPECIFIC CONCERNS

Being successful in the modeling phase is crucially important, no question about this, but it is equally important to be successful in reflecting the models in architectures that are an adequate basis for implementation. In this regard, we have identified several *domain-specific concerns*, not claiming exhaustiveness nevertheless, which are presented in the current section; those concerns are *intuitive behaviour*, *devices' technical restrictions*, *security*, and *privacy*.

3.1 Intuitive Behaviour

As mentioned before, a border police officer is sometimes especially valued for his/her capability to be intuitive with regard to a situation, applying a "sixth feeling" in deciding what to do. In our view, this is nearly impossible to capture and reflect in models. This means in turn that an important Border-Security-related "asset" would inevitably remain outside the "scope" of a CABS system.

3.2 Devices' Technical Restrictions

Often a device at the border is a piece of hardware and its electronics would often be restrictive with regard to the ways in which it could be used. If a software application would have to "bridge" the device to the system, then in-depth knowledge on the electronics of the device would be a must – this complicates the job of enterprise modelers and software designers. Those restrictions are nevertheless not only electronics-related – a drone, for example cannot "carry" more than half a kilo and cannot stay in the air more than a certain amount of time – such restrictions should be taken into account as well.

3.3 Security

A CABS system should follow highest security standards, "higher" than even in Banking because a Border Security failure may lead to dramatic consequences for hundreds and thousands of people. This assumes not only establishing advanced computer networking but also "guaranteeing" what human actors would do (or not do). This is to be taken into account and in our view, it is very difficult to actually establish and maintain so high security standards in two perspectives - personal and technical.

3.4 Privacy

A CABS system should be a privacy-sensitive system because of a number of privacy-related risks at the border, concerning both border police officers and

211

persons crossing the border. For example, it should not be possible that terrorists whose crossing the border was obstacled by a border police officer, are able to later on identify the police officer. Another example: it should not be possible that crowd monitoring information is used later on in another context, with regard to the monitored person(s). Those issues are "burdened" with many legal aspects and what makes things even more complicated is that legislation differs from country to country, even inside the EU.

Those are four important concerns, related to the application of the CABS models and they indicate that there are issues beyond enterprise modeling and software design, that need to be taken into account and have great potential impact on the CABS system –to-be.

4 CONCLUSIONS

There is room for improving both the effectiveness and efficiency of the control at the external EU borders, and context awareness is a desired feature in this regard. Nevertheless, the development of context-aware Border Security systems is not trivial because of numerous possible-to-occur situations and prediction difficulties, and because of person-specific (intuitive) and device-specific (algorithmic) behaviour patterns. A "middle-out" realization of the SDBC Approach is proposed to tackle this, coming through the derivation of a general model, a conceptual model (split in two – a humans model (HM) and an IT model (ITM)), modeling elaborations, a synchronization between HM and ITM, and enrichment in terms of data analytics. Persons' intuitive behaviour, the technical restrictions of devices, security, and privacy are among the concerns related to the application of those models. Inspired by the aim of furthering this research, we would stick to the following recommendations that we identified accordingly:

- A CABS system should be modeled as a human-centric system because the intuitive behaviour of border police officers is and will be of great importance at the border.
- In using devices, the quality-of-data is to be of great importance for a CABS system, and this issue is to be also handled functionally.
- A CABS system is to be modeled as a context-aware system where the delivered system behaviour depends on the context situation at hand.
- A CABS system is to be based on an enterprise model that is split in two – one part featuring persons and another part featuring devices; a synchronization between the two is essential.
- Security and privacy are issues that are to be taken into account additionally, in order to have a CABS system of real value.

REFERENCES

AWARENESS, 2006, the website on the Freeband AWARENESS Project: https://www.utwente.nl /ctit/research/research_projects/concluded/bsik/ freeband/projects/awareness

Dietz, J.L.G., 2006. *Enterprise Ontology, Theory and Methodology*. Springer-Verlag, Berlin Heidelberg.

FRONTEX, 2016, the website on the European Agency FRONTEX: http://frontex.europa.eu.

LandBorderSurveillance, 2012, the website on the EBF LandBorderSurveillance Project: http://ec.europa.eu/ dgs/homeaffairs/financing/fundings/projects/ project_example_048_en.htm

Liu, K., 2000. *Semiotics in Information Systems Engineering*. Cambridge University Press, Cambridge.

PERSEUS, 2015, the website on the European FP7 PERSEUS Project: http://cordis.europa.eu/ project/rcn/97515_en.html

Schmidt, D.C., 2006. *Model-Driven Engineering*. IEEE Computer Society.

Shishkov, B., 2016. Foreword. In *BMSD'16, 6th International Symposium on Business Modeling and Software Design*. SCITEPRESS.

Shishkov, B., 2005. *Software Specification Based on Re-usable Business Components (PhD Thesis)*, TU Delft – SIKS Publishing. Delft.

Shishkov, B., Dietz, J.L.G., Liu, K., 2006. Bridging the Language-Action Perspective and Organizational Semiotics in SDBC. In *ICEIS'06, 8th International Conference on Enterprise Information Systems*. SCITEPRESS.

Shishkov, B. and Janssen, M., 2016. Towards a Service-Oriented Architecture for eVoting. In *BMSD'16, 6th International Symposium on Business Modeling and Software Design*. SCITEPRESS.

Van der Aalst, W., 2011. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin Heidelberg.

212