UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Zaitsev Artem

Comparison of STS and ArchiMate Risk and Security Overlay

Master's Thesis (30 ECTS)

Supervisor(s): Raimundas Matulevičius

Tartu 2018

Comparison of STS and ArchiMate Risk and Security Overlay

Abstract:

Nowadays ArchiMate is widely used in enterprise architecture modelling of the various business domains and briefly could be described as something in between UML and BPMN with main focus in architectural perspective. STS in its turn is focusing on socio-technical perspective and taking into consideration social interactions betwen actors. Current state of the art is talking about Secure Socio-Technical Systems and ArchiMate separately. This is perfectly fine because this two approaches are quite different. Still, they have a lot in common. Based on the state described above problem could be identified as an absence of tools or approaches which will combine these two approaches into a new one, which will take into consideration both architectural and socio-technical perspectives of modelling. This combination could be beneficial because ArchiMate risk and security overlay models risk management and STS models how actors involved in this system interact with each other from social point of view and highlights “human factor” in security. Thus, combination of them could potentially result in security modelling approach which will cover both architecture and social points of view. Ideally, this approach will create some workarounds over weak places in both initial approaches and heavily use their best parts. We will also validate this approach in terms of completeness with respect to ISSRM. In this paper we will describe this combined approach.

Keywords: STS, ArchiMate, ISSRM, Security requirements engineering, Security risk management, ArchiMate Risk and Security Overlay

CERCS: P170 Computer science, numerical analysis, systems, control

STSi ja ArchiMate'i võrdlus turvalisuse modelleerimisel

Lühikokkuvõte:

ArchiMate'i kasutatakse tänapäeval laialdaselt erinevates ärivaldkondades ettevõttesüsteemide arhitektuuri modelleerimiseks ning seda võib iseloomustada modelleerimise tööriistana, mis ühendab endas UML'i ja BPMN'i. STS keskendub aga sotsiotehnilisele perspektiivile ja tegijatevahelistele sotsiaalsetele vastastikmõjudele. Kuigi neil on palju ühist, on tegemist siiski erinevate lähenemistega, mistõttu räägitakse tänapäeval ArchiMate'st ja Secure Socio-Technical Systems'ist valdavalt kui eraldiseisvatest süsteemidest. Sellise olukorra tõttu on tekkinud puudujääk tööriistadest ja lähenemistest, mis ühendaks kaks süsteemi üheks uueks, mis võtaks arvesse nii modelleerimise arhitektuurseid kui ka sotsiotehnilisi aspekte. Selline kombinatsioon võib osutuda kasulikuks, kuna ArchiMate'ga saab modelleerida riskijuhtimist ja STS abil saab modelleerida erinevate süsteemi kaasatud tegijate omavahelist suhtlemist sotsiaalsest vaatevinklist ja turvalisuse inimfaktorit. Seega nende kahe süsteemi ühendamise teel võib luua turvalisuse modelleerimise lähenemise, mis katab nii arhitektuurilised kui sotsiaalsed vaatevinklid. Ideaalselt kasutaks selline lähenemine mõlema süsteemi tugevamaid külgi ja lahendaks mõned kitsaskohad. Lähenemise terviklikust hinnatakse ISSRM'i suhtes. Selles lõputöös kirjeldatakse ülalmainitud kombineeritud lähenemist turvalisuse modelleerimisele.

Võtmesõnad: STS, ArchiMate, ISSRM, Turvanõuete insener, Turvariski juhtimine, ArchiMate risk and Security Overlay

CERCS: P170 Arvutiteadus, arvanalüüs, süsteemid, kontroll

List of Abbreviations

List of Abbreviations that could be found in this paper.

ISSRM	Information system security risk management
STS	Secure socio-technical systems
ArchiMate	Architecture-Animate
STS-ml	STS modelling language
SRE	Security Requirements Engineering
ERM	Enterprise Risk Management
RSO	Risk and Security Overlay
BTC	Blood Transfusion Centre
RSO	Risk and Security Overlay
SRM	Security risk management
UML	Unified modelling language
BPMN	Business process model and notation
BPM	Business process model
DEMO	Design and Engineering Methodology for Organizations

Table of Contents

List of Tables

List of Figures

1 Introduction

Nowadays software has a huge influence on all aspects of our life. From internet surfing and talking with someone online to bank transactions and social networks. Huge amount of people's personal data goes through software systems and no one wants it to be stolen. Moreover, people willing to use all this system without any errors. Therefore, from day to day software security becomes more and more important in real life. Swift growth of distributed systems forced developers, designers, managers and all other people related to developing software to consider software security as one of the most important activities in developing cycle.

Software security does not depend only on outer reasons such as firewalls, but also on inner software application security. Inner security is a main anxiety for modern software systems and it could be achieved by system modelling in context of threats and security.

Huge variety of different modelling approaches has been developed to solve the software security issues. In this paper I will take a closer look on STS method of designing secure software systems and ArchiMate - modelling language for enterprise architecture. After describing in details what are the main things in this approaches we will compare and integrate them. Integration will be validated either by the proof of concept (prototype) or by illustrating the usage on extensive example.

Scope of this work will mainly be in security risk managementfield. In this field STS and ArchiMate risk and security overlay were chosen to work with. Note, that we are taking into consideration only risk and security overlay, not the whole core ArchiMate.

Regarding motivation: Combining ArchiMate and STS could be a good idea because they both are visual notations for modelling security risks and their main focus are slightly different. Main idea of STS is that designing secure software systems should take into consideration socio-technical perspective of the system along with technical side instead of modelling only technical side. ArchiMate, in its turn, is all about architectural modelling (includes business, technical and infrastructural architectures). Thus, combination of them could easily benefit from ArchiMate in architectural perspective and from STS in socio-technical perspective, leading into potentially good way to model security risks.

Research problem is mainly about integrating ArchiMate and STS. State of the art shows that there are no integrations of this two approaches, therefore we need to find out how to integrate ArchiMate and STS into one approach and how to evaluate integrated approach.

1.1 Research Question

Ergo, main research question of this paper could be formulated as following:

How could ArchiMate and STS be integrated into one security risk modelling approach, which will combine advantages from each of them?

Defined sub-questions are following:

What is ArchiMate risk and security overlay and how it maps into ISSRM?

What is STS and how it maps into ISSRM?

What is comparison criteria for ArchiMate RSO and STS?

What is the optimal way to integrate ArchiMate RSO and STS?

How can integrated approach benefit from ArchiMate RSO and STS?

How can integrated approach be evaluated?

What is completeness of integrated approach regarding ISSRM?

What are the directions of future work to improve integrated approach?

1.2 Summary of the contribution

Main contribution consists from presenting new risk and security management modelling approach which was created from integration of STS and ArchiMate RSO. To achieve integration of STS and ArchiMate RSO we first compared them under ISSRM terms. To be able to compare STS and RSO under ISSRM concepts we initially mapped RSO and STS concepts to ISSRM concepts. Integrated approach is built in the following way: we took STS as "skeleton" for new approach and noticed that STS lacks a lot of concepts in risk and security field. STS has only "threat event" and relationship "threatens" to model threats and risks. That's why we introduced whole new view and called it "risk and security view". We populated this view with RSO models which are connected to threat events on social and information views (we allowed to use threat events on information views to better model threat which are not social based). New integrated approach is presented with expanded explanation which consists from semantics, concrete syntax and application guidelines for it. We decided to evaluate integrated approach in terms of completeness for ISSRM concepts. We calculated completeness by calculation how much of ISSRM concepts (elements and relationships) are presented in approach. To be able to compare results for integrated approach with other approaches we selected two additional approaches (besides STS, RSO and integrated): Secure Tropos, because it correlates with STS and DEMO, because it correlates with ArchiMate. Results showed that integrated approach is the only one among compared approaches that fully covers ISSRM concepts, thus making it 100% complete in ISSRM terms. In addition, we modelled same case through five different approaches and compared outcome models. This comparison highlighted good sides of integrated approach (pointed out on various places where considered attack is possible, heavy loaded with information risk and security view) and downsides too (the most complex and most time-consuming model building among all compared approaches).

1.3 Case for application guidelines

As for example of usage I will use healthcare scenario derived from Hong Kong Red Cross Blood Transfusion Centre (Red Cross BTC). Detailed description of case is available at Hong Kong Red Cross BTC website[1]. I was influenced by STS book [1], as it is used there.

In this example organisations (hospitals, labs) interact with each other and with humans (patients, doctors, nurses) to provide healthcare services. Doctors use medical equipment to care about patients, patients provide their personal data in order to receive healthcare, hospital personnel enter patient personal data into information system and so on.

Alice is one of the donors who periodically donates blood. Centre takes responsibility of collecting and testing blood. It is also responsible for transferring applicable blood to different hospitals. Red Cross BTC also should check eligibility of donors, but the actual blood tests are done in special laboratories.

Hospitals hold all info about their patients along with info about transfusions. Information about transfusion, reason for it, type of operation, medical personnel etc. are stored in hospital database. Actual healthcare services are provided to patients by physicians, who got access to patient personal information.

[1] https://www5.ha.org.hk/rcbts/enindex.asp

Patients personal info and blood consumption rates can be accessed by Red Cross BTC to perform statistical and analysis estimate blood consumption rates in future and to make recommendations on blood usage.

Hospital authority setups privacy regulations concerning patient’s personal data. Red Cross BTC reports to hospital authority, which keeps patient’s information private according to regulations.

Overall success of this system heavily depends on efficient interaction between physicians, labs, patients, hospitals, hospital authority, donors and Red Cross BTC. Each actor has its own security expectations, what limits the way how other actors can interact with assets that they want to protect. For example, donors allow Red Cross BTC to collect their personal data and use it for statistical analysis but they don't want to expose this information to third parties. In case if a famous people hospitalised they prefer all information related to their conditions to be confidential.

2 Security risk management

This chapter covers security risk management frameworks, such as ISSRM, Cybersecurity framework and Practical information risk management process framework. Summary provides argumentation why ISSRM was chosen among other options.

2.1 What is information system security risk management (ISSRM)?

ISSRM stands for information systems security risk management, and described in [9]. ISSRM was developed by a survey of security risk management methods and security risk management standards and summarises in ISSRM domain model represented below. In a few words - ISSRM is a framework to compare and standardise other security risk management approaches.

ISSRM domain model consist from three main aspects: risk treatment-related concepts, risk-related concepts and asset-related concepts. In general domain model represented as UML diagram below. This figure is taken from [9].

Asset-related concepts show which of organization's assets are important and requires protection. Asset - anything what is valuable for organisation and what is needed to reach organization's goals. Assets divides into business and IS assets. Business assets represents skills, processes, capabilities and information important for business core functionality. IS assets represents parts/components of information system which are important for organisation since they support business assets. IS assets are usually part of information system (hardware, software) or person/facility who plays role in a system. Security constraints represents security need and usually are properties of business assets [9].

Risk-related concepts represent risk itself and its components. Risk is a combination of threat and vulnerability which leads to damage for two or more assets, what is called impact. Impact is a potential negative effects on business assets. Impact could be defined at IS asset level (broken component), or at business asset level where it brakes security criteria (loss of availability, loss of confidentiality).

Risk event - combination of threat and several vulnerabilities. Vulnerability - property of IS asset that represents weakness in security. Threat - incident done by threat agent that exploits vulnerabilities and causes damage to assets. Threat agent - agent who intentionally want to harm assets [9].

Risk-treatment related concepts represent concepts to treat risk. Risk treatment decision - decision on how to treat identified risk, which satisfies security need. There are four types of decisions:

- Risk avoidance: decision to avoid risk, performed by withdrawing or modifying part of functionality.
- Risk reduction: actions to make risk less likely to happen, reduce risk impact.
- Risk transfer: decisions related to how parties share damage from risk.
- Risk retention: compose acceptance of damage from the risk.

Security requirement - environment condition that we bring to life by using information system in order to soften risks. Control (countermeasure, safeguard) is created to implement security requirements. Security controls can be represented by devices, policies, processes, or other components of IS asset that act to reduce risk [9].

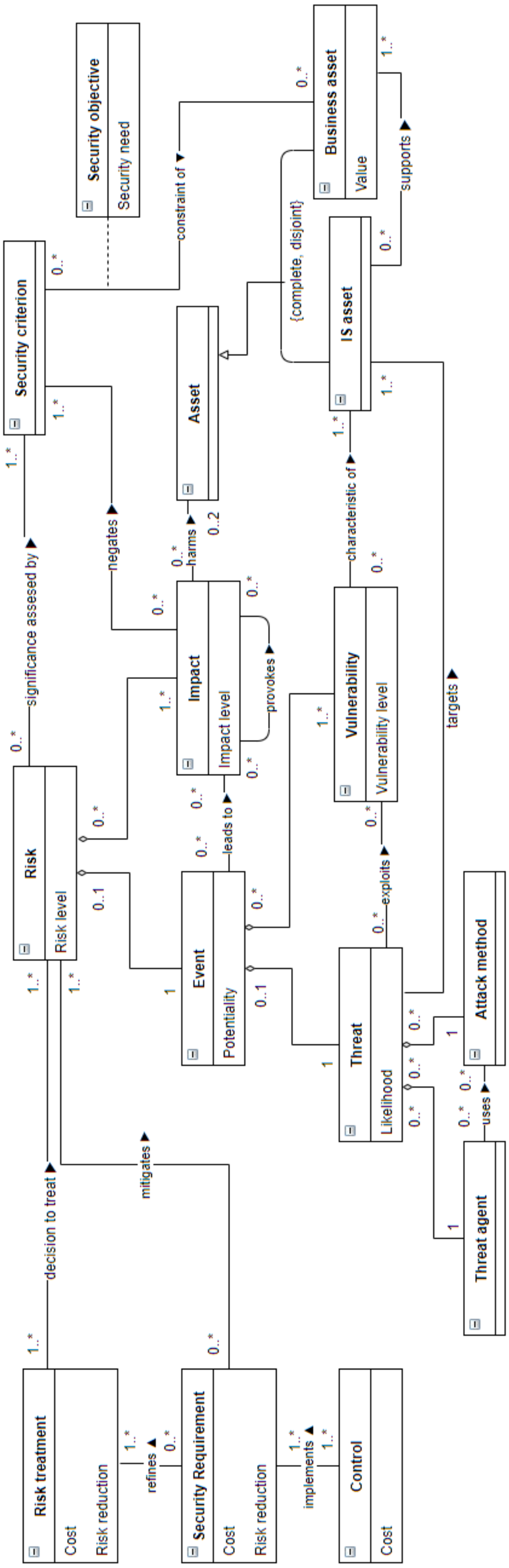


Figure 1. ISSRM domain model [9] [2]

[2] This and following models with same style are built in online tool available at: https://www.draw.io/

Regarding relationships in ISSRM. IS asset can *support* business asset in any form, but business asset can be standalone, without any support from IS assets (employee skills). Security criterion can *constraint* some or none business assets, but business assets not necessarily need to be *constrained* by security criteria. Event should always *lead to* impact, if there is no impact – event doesn't make any risk. Impact could *provoke* other unpredicted impacts. Impact always *harms* assets; however, asset could remain unharmed. Also, impact always *negates* at least one security criterion, but security criterion could not be *negated by* impacts. Threat could *exploit* some vulnerabilities or none of them. Vulnerability can be *exploited by* none or several threats. Vulnerability is always a *characteristic of* asset, but asset could not have any vulnerabilities. Threat always *targets* one or more assets and asset can be not *targeted*. Threat agent could *use* attack method, and one attack method could be *used* by several threat agents. Risk treatment always points to *decision to treat* at least one risk. Risk treatment could be *refined* to zero or more security requirements, but security requirements always *refines* one or more risk treatments. Security requirement always *mitigates* at least one risk, but risk could not have related security requirements. Control always *implements* one or more security requirements, and security requirements is always *implemented by* at least one control.

2.2 Cybersecurity framework

Cybersecurity framework is a risk-based approach to manage cybersecurity risks in information systems and is developed in National Institute of Standards and Technology, United States Department of Commerce. Latest framework version is described in [18], following framework description is based on this whitepaper. Framework consist from three parts: Core, Implementation Tiers, Profiles. Each framework part consolidates relations between business drivers and cybersecurity activities. *Core* framework includes cybersecurity activities, outcomes and references, which are widely used in sector. It presents common guidelines, best practices and industry standards. Core part consist from five functions - Identify, Protect, Detect, Respond, Recover – which together provide strategic high-level view on organizations security risk management. After functions give their outcomes, framework maps this results to existing standarts, guidelines or practices for each category. *Implementation tiers* gives context about organization's views on risks and processes to manage them. They also state how well organization risk management are compared to the ones described in framework. Tiers characterize organization's processes from partial (tier 1) to adaptive (tier 4). These tiers show company's progress towards agile and risk-informed processes. *Framework profile* shows outcomes from business needs, which are selected from framework categories. Overall, profile is a combination of practices, standards and guidelines. They are used to compare "as is" profile with "to be" profile. To create a profile, organization review all their processes along with categories and subcategories from the point of view of business drivers and risk assessment. Current organization profile used for prioritization and following progress to "to be" profile.

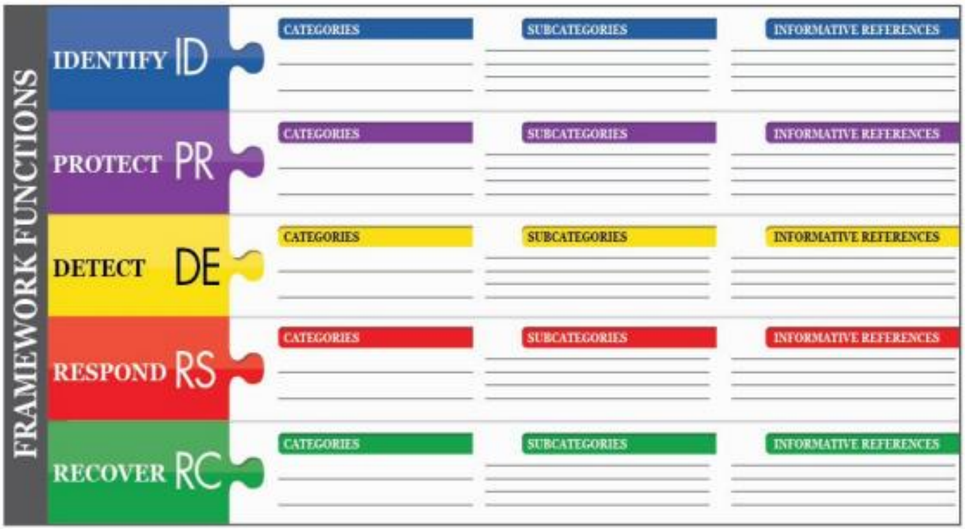


Figure 2. Cybersecurity framework core structure [18]

2.3 Practical Information Risk Management Process Framework

This framework is described in [19]. Following description is based on this paper. It is focusing on managing information area of risks. Framework consist from following steps: Identify (identify event which can cause risk), Analyze (measurements and prioritization of threats to select information security controls), Control (anything that mitigates threats and vulnerabilities), Monitor (process of systematical observing information controls performance), Report (systematic reporting to decision makers). These steps applied for the following domains: Strategic (executive level, high-level objectives, policies and plans), Tactical (management activities at program level, application of best-practices that meets long-term and short-term objectives), Operational (low-level processes which instantly affects everyday operations). Application of steps on domains from the strategic tactical or operational point of view is all what this framework is about.

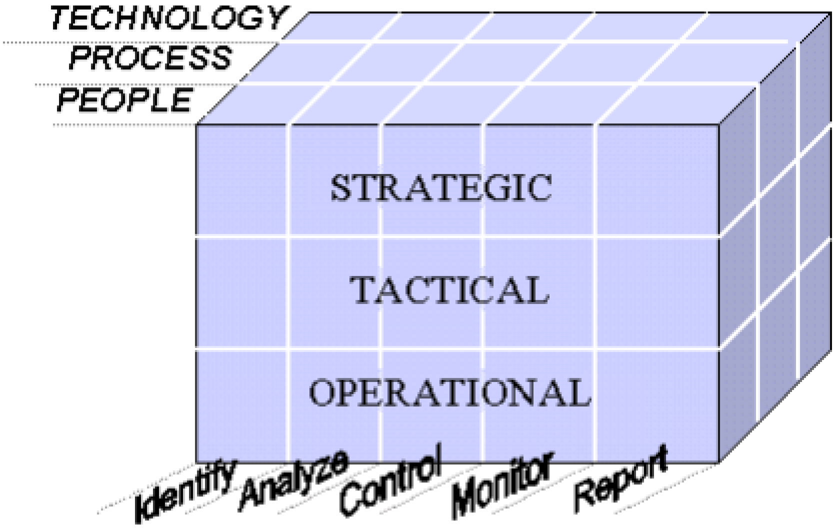


Figure 3. Information Risk Management Process Framework [19]

2.4 Summary

This chapter briefly describes three frameworks for security risk management. Practical information risk management process framework and Cybersecurity frameworks provide best practices and tools to analyse risks with aim to come up with risk controls, but both of them lack a structured and described domain model. We need domain model to have a tool for mapping modelling languages and their comparison in order to be able to integrate them. Therefore, we choose ISSRM as our security risk management tool for this paper. Based on ISSRM and its domain model we will make comparison between STS and ArchiMate RSO and integrate them.

3 Modelling languages

3.1 Security and security risk management modelling languages

This section will provide brief overview of security and risk management modelling languages. We will not pay much attention to this languages, as our main focus remains on STS and ArchiMate RSO.

First modelling language that we will talk about is security risk-aware Secure Tropos. We decided to talk about it as it relates to STS. They both pay attention to social perspective of the system. This approach is introduced by Mouratidis in [13, 14, 15]. This approach is based on Tropos methodology [16]. Secure Tropos define system development through following phases: Early requirements (helps to define problem), Late requirements (defines desired state of system), Architectural design (deals with system architecture), Detailed design (defines each component by input, output, controls). Secure Tropos is separated by perspective of several models.

Actor model defines actors and their dependency relationships. To enforce security limitations, language introduce security limitations. They settle up security restrictions that actors must respect and system must follow. Dependency shows that one actor depends on another for any reason. Same goes for security dependencies, but they show that actors depend on each other to meet security limitations.

Goal model represents actor's reasons for fulfilling goals, plans and resources. Hardgoals represent main interests. Softgoals, in its turn, doesn't have clear satisfaction criteria and depends on interpretation. Plan is a plan of how to do stuff. Resources represents valuable entity. Also, Means-Ends, Contribution and decomposition relationships are used on this model. Secure goals are same as regular goals, but they focus on security. Achievement of secure goal is described in secure task. There are also security reference diagrams, they represent conditions that can endanger system's security features. Actual attack is defined by security attack scenario. Modelers can specify attacker's role and his goals, and try to guess possible attacks on the system.

Secure Tropos constructs are shown on the figure below, which is adapted from [9].

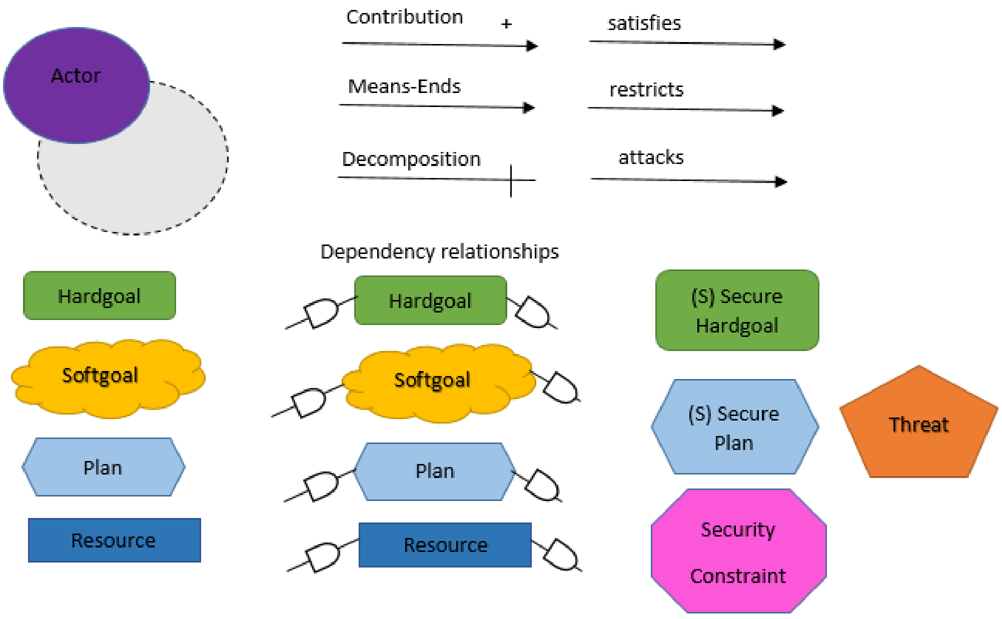


Figure 4. Secure Tropos language constructs [9]

Second chosen modelling language is Design and Engineering Methodology for Organizations (DEMO). We choose it as a counterpart for ArchiMate RSO, as they both are designed for enterprise modelling. First of all, DEMO is an enterprise modeling methodology, with a main idea of "communicative action". It considers communication as essential part of functioning organization [20]. DEMO relies on DEMO specification language (DEMOSL) to graphically represent DEMO models. DEMOSL is described mainly in [21]. Following description is based on [21]. To model organization, DEMO build four models, which models different view on organization. *Construction model* gives view on organizations construction: composition (actors inside organization), environment (actors outside from organization), interaction structure (interactions between composition and environment), interstriction structure (information links between composition and environment). *Action model* gives view on organizations operations. This one consist from sets of action rules and work instructions. Action rule describes what should be done. Work instruction is not necessary; they mainly guide on how to perform action rule. *Process model* gives view on organization's state space and transition space from perspective of coordination. It also contains entire process steps and transaction pattern, along with occurrence laws between transactions, which are visualized in links between process steps. *Fact model* gives view on organization's state space and transition space from perspective of production. It contains identified facts and existence laws (reference, unicity and dependency are visualized). This model also contains production event and occurrence laws for them. As scope of this work is in security and risk management, we will pay attention only to construction model, because only it is relevant for current scope, and it only makes sense to model violator on it (because it is the only place where we can model actors outside of organization).

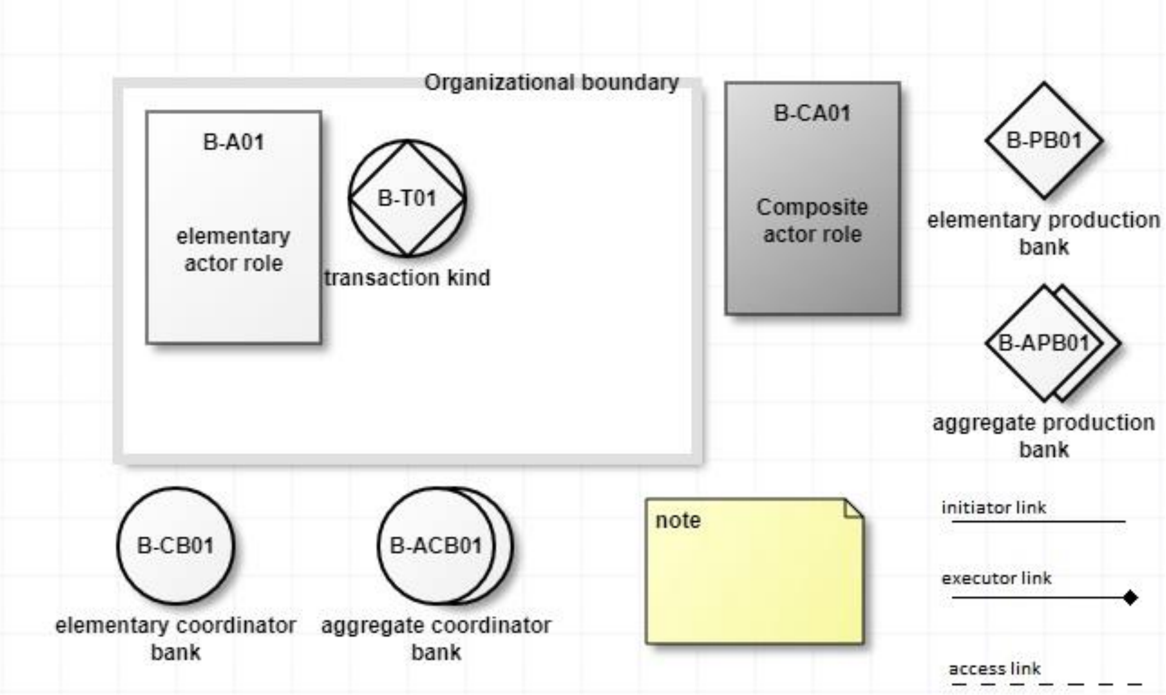


Figure 5. DEMO construction model elements

3.2 What is Secure Socio-Technical Systems (STS)

STS is a method for designing secure software systems [1]. This method heavily focuses on the first stages of designing the system - requirements engineering. Main idea behind STS is that design of the secure software system should take into consideration socio-technical perspective of the system along with technical aspects of the system. STS is model-driven which means that main activity in it is constructing a model. Constructed model represents security requirements of the system. This model is build using STS modelling language (STS-ml). This modelling language is the most important part of the model-driven way in SRE proposed by STS. STS-ml follows ten principles to ensure that if follows requirements for modelling languages and that it fits into socio-technical security requirements domain [1]. These principles are described below.

Principle 1: Socio-Technical perspective. Modelling primitives must be enough to create security requirements models for socio-technical systems which involves humans, organizations and software systems. These actors should interact with each other to reach their strategic objectives and fulfil requirements.

Principle 2: Security about interactions. All participants in socio-technical systems are fully autonomous, therefore they are not controllable. Still, while interacting with other actors (to reach goals or transfer information), one actor usually imposes security requirements on another one.

Principle 3: High-level assets. Modelling language should give opportunity to model high-level assets of actors (objectives or information).

Principle 4: Threats. Modelling language should provide functionality to describe and identify social and organizational threats, which are not related to technical vulnerabilities.

Principle 5: Multiple Stakeholders. Each stakeholder has his own security requirements and sometimes they could conflict with another stakeholder security requirement. Therefore, language should capture stakeholder's viewpoints and help with identification of conflicts.

Principle 6: Diagrammatic and formal language. Formal semantics through primitives should be provided and language must support modelling through diagrams.

Principle 7: Compliance with standards. When it is possible primitives should refer to the same standardised terminology.

Principle 8: Minimality of concepts. Language should contain minimum set of primitives that are needed to capture security requirements.

Principle 9: Traceability. Security requirements should be traceable to requester and his goals which caused this requirement to appear.

Principle 10: Capturing security needs. Language should focus on needs of stakeholders and not on solutions to provide that needs.

Main goal of STS-ml is to represent and visualise security requirements of system. Following principle 7, stated at the beginning of this chapter, STS-ml should follow well know terminology. Unfortunately, there is no consensus on what basic terminology for SRE should be. Therefore, STS-ml proposes classification which consist of the following aspects: *confidentiality, integrity, availability, authenticity, reliability* and *accountability.* In the following text will be explained how STS-ml fulfils each of this aspects.

Confidentiality. This aspect represents the idea that confidential information should not be available for unauthorised users. In STS-ml confidentiality requirements represented by different types of authorisations.

Integrity. This requirement ensures that information is not modified or deleted in unauthorised way. In STS-ml integrity requirements for modification represented by authorisation type with specified Modify rules. Unfortunately, it is very challenging to capture deletion of information, because information could be tangible by various documents and their copies, thus STS-ml does not focus on it.

Availability. This requirement means that system uptime is relatively high and system did not deny authorised users, and provides access to requested information in relatively good time. STS-ml differentiate between document availability and Goal availability. Document availability means that actor who provides document provide it with certain level of availability for actor who requests this document. It is usually visualized over document transmission with a sticky note with required percentage of availability. Goal availability means exactly the same as Document availability but in context of goals.

Authenticity. This security aspect catches the ability to be authenticated verified and trusted. It is checked through authentication process which goal is to verify that users are who they say they are. STS-ml express authenticity over interactions between actors (goal delegations and document transmissions). Moreover, STS-ml defines authenticity in two variants: authentication of delegator/sender and authentication of delegatee/receiver.

Reliability. These security aspects represent threat from accidental errors. Usually under it security engineers mean that attackers trying to misuse system. In STS-ml reliability is represented through *trustworthiness* and *redundancy*. Trustworthiness requires delegatee to be trustworthy (provide certificate for example) and it is expressed over goal delegations by delegator. Redundancy is expressed over goal delegation and means that delegator wants delegatee to adopt redundant strategies for delegated goal by using alternative strategies (single actor) or by relying on other actors (multi-actor).

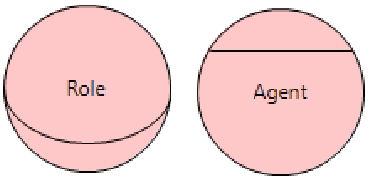
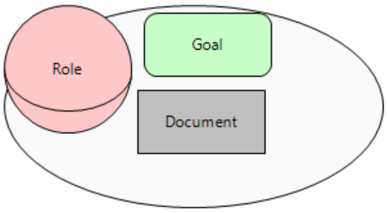
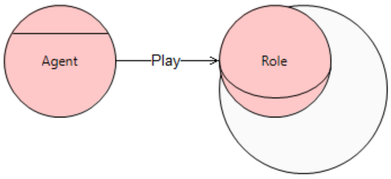
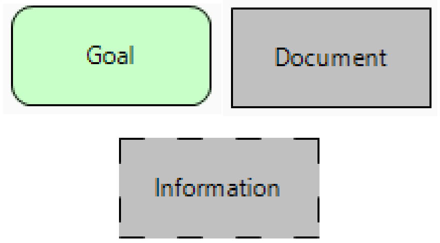
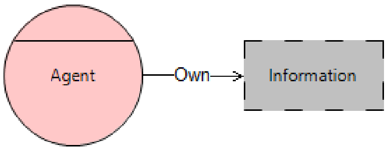
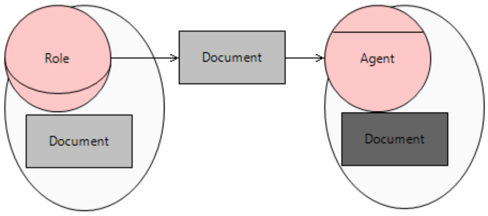
Accountability. These aspects refer to the requirements for actions of an entity to be traced uniquely to that entity [2]. STS-ml supports this aspect in the following ways: *non-reduplication, non-redelegation, separation of duties, combination of duties*. Non-reduplication means that sender or receiver should not be able to deny a transmitted message. Non-redelegation means that delegator request delegatee to take full responsibility for achieving delegated goal on himself, without any other actors. Separation of duties means separation of duties among different people when they are dealing with critical tasks (for example - opening bank deposit box requires effort and information from different people). Combination of duties (sometimes in SRE literature called *retain familiar* [3]) means that same entities will be assigned to the same actor.

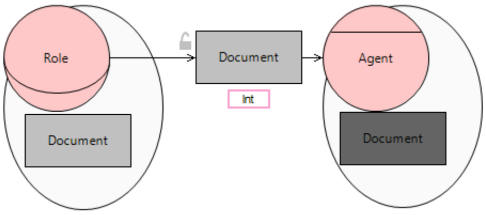
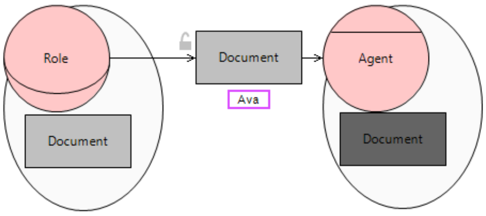
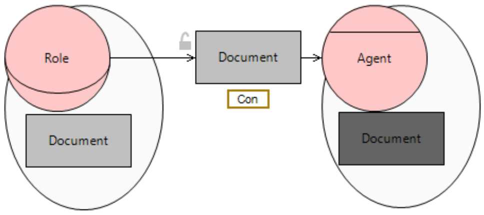
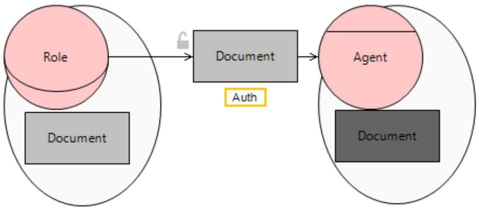
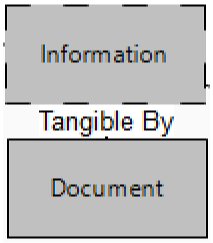
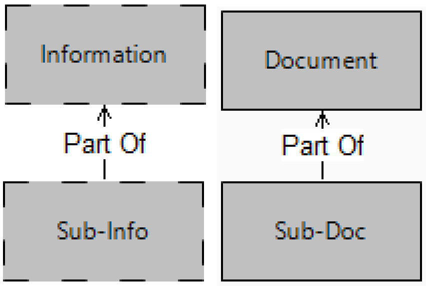
All security requirements described above are modelled through rectangle notation with first few letters of the aspect putted over the related entity in the model. If modelling requires additional information - it is usually visualised as sticky note near the rectangle notation.

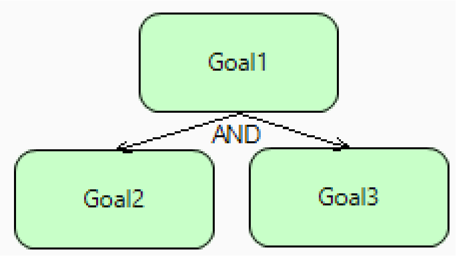
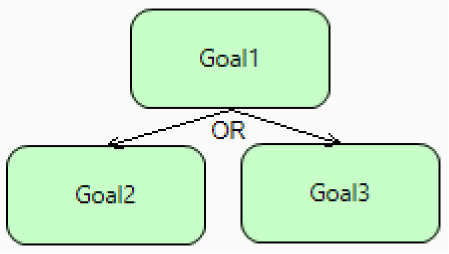
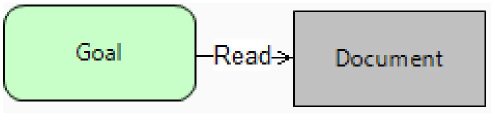
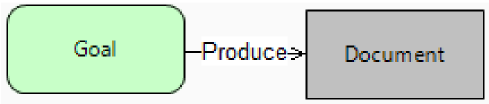
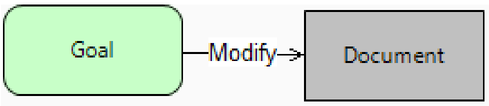
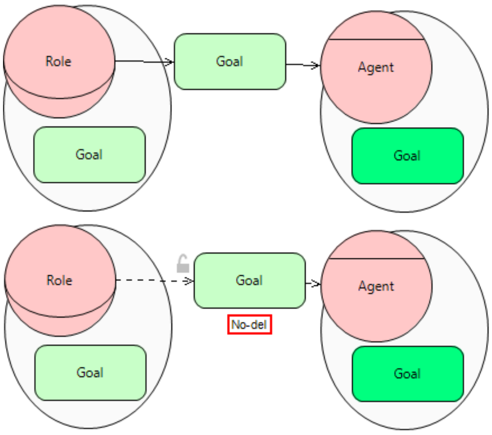
3.2.1 Semantics and Concrete syntax

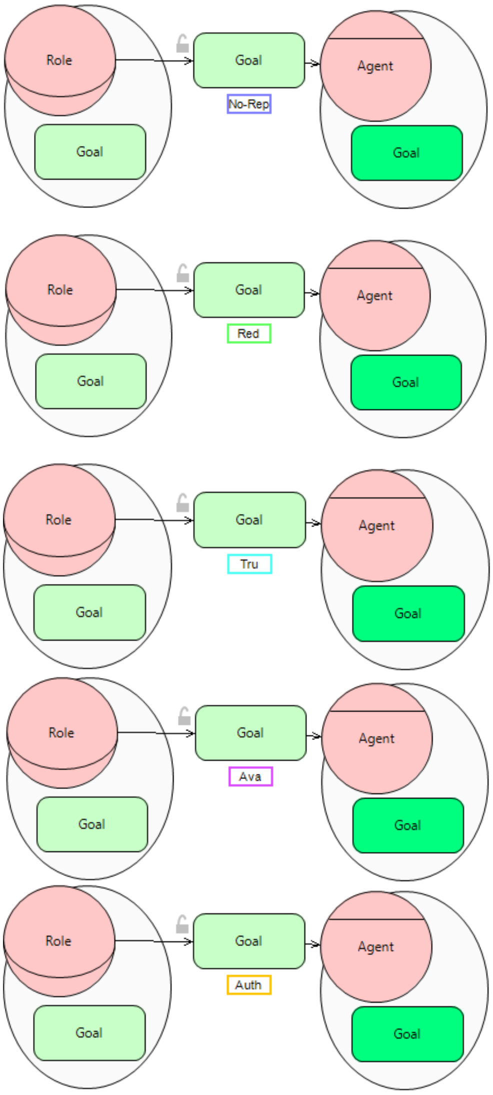
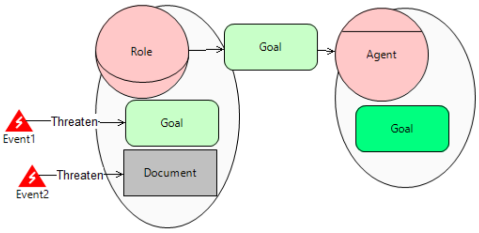
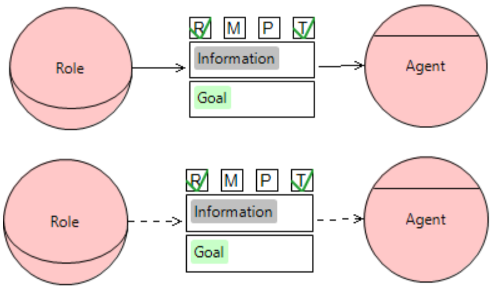
Following table presents semantics and concrete syntax for STS. It makes perfect sense to combine semantics and concrete syntax in one table because we show how element looks like and explains meaning behind it in one place, which makes semantics and concrete syntax easier to remember and understand.

Table 1. STS-ml semantics and concrete syntax [8]

Concept	Graphical Notation	Explanation
Role/Agent		Role is an abstraction of the actor (student). Agent represents concrete actor (John).
Role/Agent Scope **Possess** **Want**		Role/Agent scope represents which goals this role/agent want to achieve and which documents does this role/agent possess.
Play		Concrete agent plays role (John plays role Student)
Goal **Document** **Information**		Goal represents desired state of affairs (car is bought). Document represents exchangeable thing (paper document) which usually contains information (salary).
Own		Role/Agent is a full owner of some information and can decide to transfer rights about it to other agents/roles
Transmit		Role/Agent transmits document in his possession to another role/agent.

Integrity of transmission		Role/agent requires integrity of transmission form role/agent who provides document
Availability		Requester demands minimum level of availability for document provided.
Confidentiality of transmission		Provider demands confident transmission of document to receiver
Authentication		Sender/receiver requires authentication on sending/receiving
Tangible by		Information is made tangible by representing it in a document
Part of	 Information Part Of Sub-Info Document Part Of Sub-Doc	One information is part of another information. Same with documents. (“Name” is part of “Personal Information”).

AND-decomposition		And-decomposition means that through achieving goals 2 and 3, Goal 1 will be also achieved. (sending letter -> letter written, letter sent).
OR-decomposition		Or-decomposition means that achieving either of the goals 2 or 3 yields achieving goal 1. (letter written -> new letter written, letter copied).
Read		Document should be read to achieve goal.
Produce		Document is produced while achieving goal.
Modify		Document is modified while achieving a goal.
Goal delegation **No-redelegation** **Non-repudiation** **Redundancy** **Trustworthiness** **Availability** **Authentication**	 Role Goal Goal Agent Goal Role Goal Goal No-del Agent Goal	Delegation: It means that role/agent delegates his goal to another role/agent. No-redelegation: Delegatee cannot delegate goal to third roles/agents. Non-repudiation: Delegatee cannot deny that goal was delegated Redundancy: Delegatee must take steps to ensure redundant goal fulfilment Trustworthiness: Delegatee must provide a proof of his trustworthiness

	 Role Goal Agent No-Rep Red Tru Ava Auth	Availability: Delegatee must provide required level of availability for delegated goal Authentication: Delegator/Delegatee shall authenticate before delegating/receiving goal.
Events	 Role Goal Agent Threaten Event1 Threaten Event2 Document	Event can threaten goals, goal delegations, documents and actors.
Authorization		Role/Agent authorises another role/agent to perform some operations on stated information in scope of the described goal. Operations: Read, Modify, Produce, Transmit. Scope could be one or more goals

3.2.2 Abstract syntax

Here we propose two new abstract concepts. First one - Actor includes Agent and Role elements. Actor has its name and can play Role. Second introduced abstract concept is Actor elements which are possessed by Actor. Actor elements include goal, document and information elements. Information and documents could be split into sub-info and subdocuments with part of relations. Information could be represented in documents which is visualised by tangible relation. Goal can perform actions over document to achieve current goal. Moreover, goal can be decomposed into sub-goals with and/or relations. There is also Event element which can threaten goal and document elements.

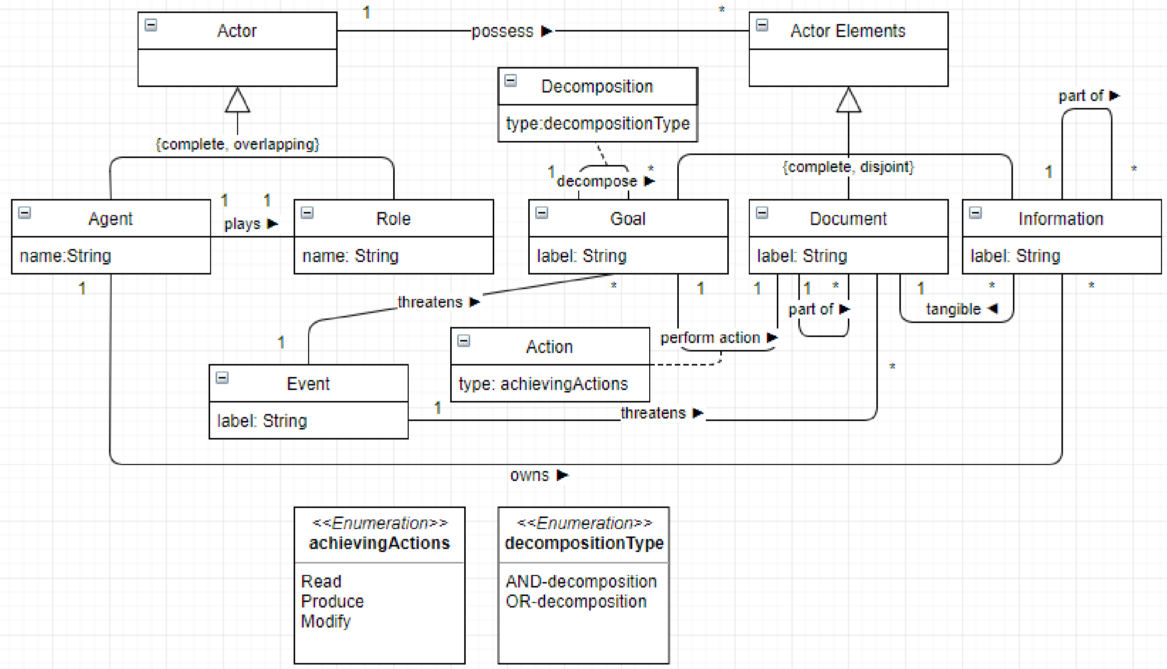


Figure 6. STS abstract syntax, main elements

Next diagram visualizes in details advanced Actor related elements in STS modelling language. These relations are document transmit, goal delegation and authorisation. Transmit relation is represented by transmit rules and relation to document element. Goal delegation element is represented by various delegation type rules and relation to goal element. Authorisation element is represented by authorization rules enumeration and set scope relation with goal along with on relations with document and information.

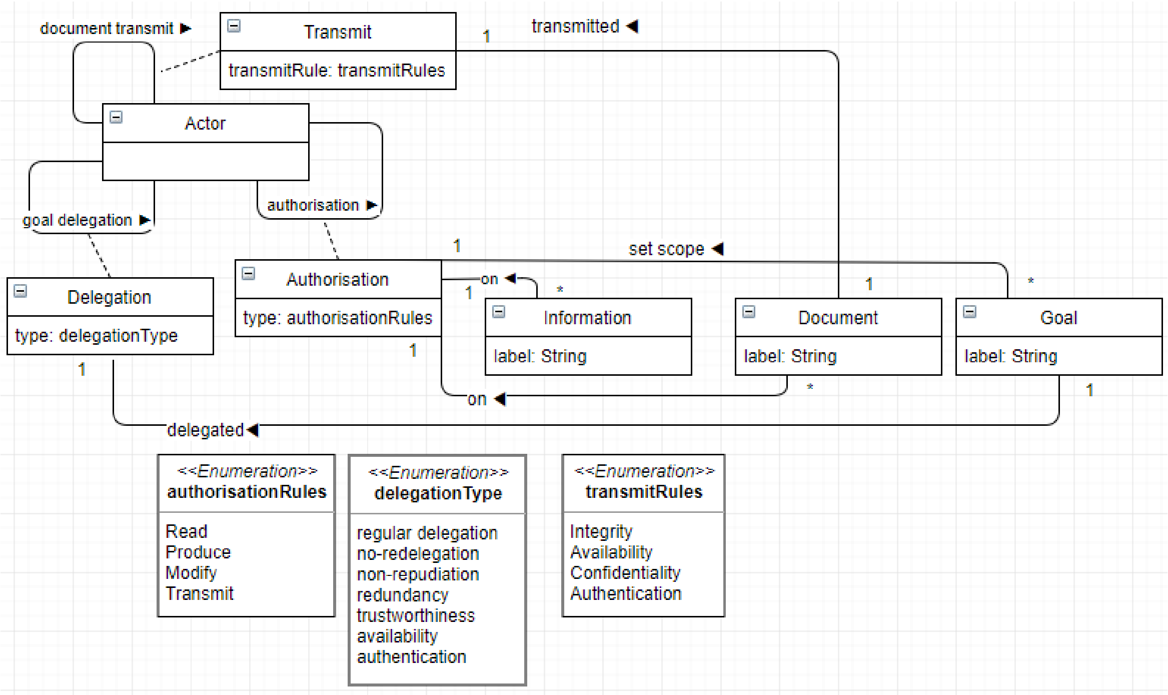


Figure 7. STS abstract syntax, advanced elements

3.2.3 Application guidelines

This chapter will contain example of modelled healthcare scenario which is described above, along detailed guides on how to use STS to model particular examples from healthcare scenario. In the end of the chapter there will be diagrams for complete scenario. Let's start from information view and in detail discuss case about Alice as a donor. In scope of information view, we have following information about Alice. She owns her personal information and health status which we will visualise with *own* relationship between Alice and this information. Both this pieces of information forms Alice's medical history, therefore we will create *part of* relations between new information entity *medical history info* and her personal information with health status. In STS terms if we want to use information somehow it must be represented in documents. From scenario scope we can say that Alice's health status and personal info are represented in her *test results* (we use *tangible by* relation to model this). Moreover, Alice's personal info must be represented in her donor certificate and in hospitals health record. We also know that test result belongs to Laboratory which makes tests, and health record with donor certificate belongs to blood transfusion centre. Diagram for this case is presented below.

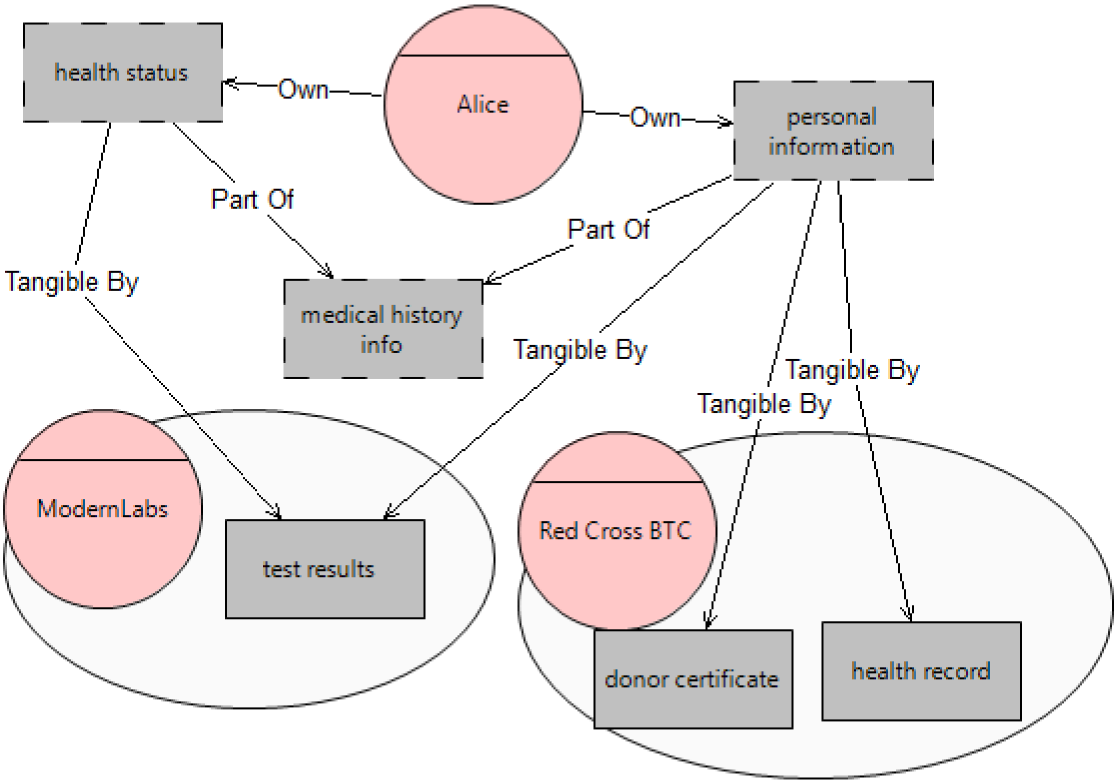


Figure 8. STS information view Alice example[3]

For authorization view let's take Alice's example too. Alice's owning of her personal info and health status along with representing it in medical history info remain the same as in information view. It is relevant to show which Role does Alice play in this view, so we show that she plays Donor role with *play* relationship. We know from previous example that Laboratory uses Alice's health status and personal info to create document test results. This means that Alice must authorise Laboratory to have access to information that she owns. ModernLabs need to read and produce this info, and we model it with green checkmarks on related authorization panel. Also, it is necessary to sat that Alice allows ModernLabs to read and produce her info only for providing test results, that's why we set scope for goal results provided in this authorization. We also know that blood transfusion centre requires Alice's health status to approve hear as a donor, and Alice allows BTC to read but prohibits to modify her health status. This information is modelled in authorization element between Alice and Red Cross BTC. Also, Alice must authorize role physician to read and produce her health status. Diagram for this case presented below.

[3] This and following STS models build in STS tool available at: http://www.sts-tool.eu/downloads/

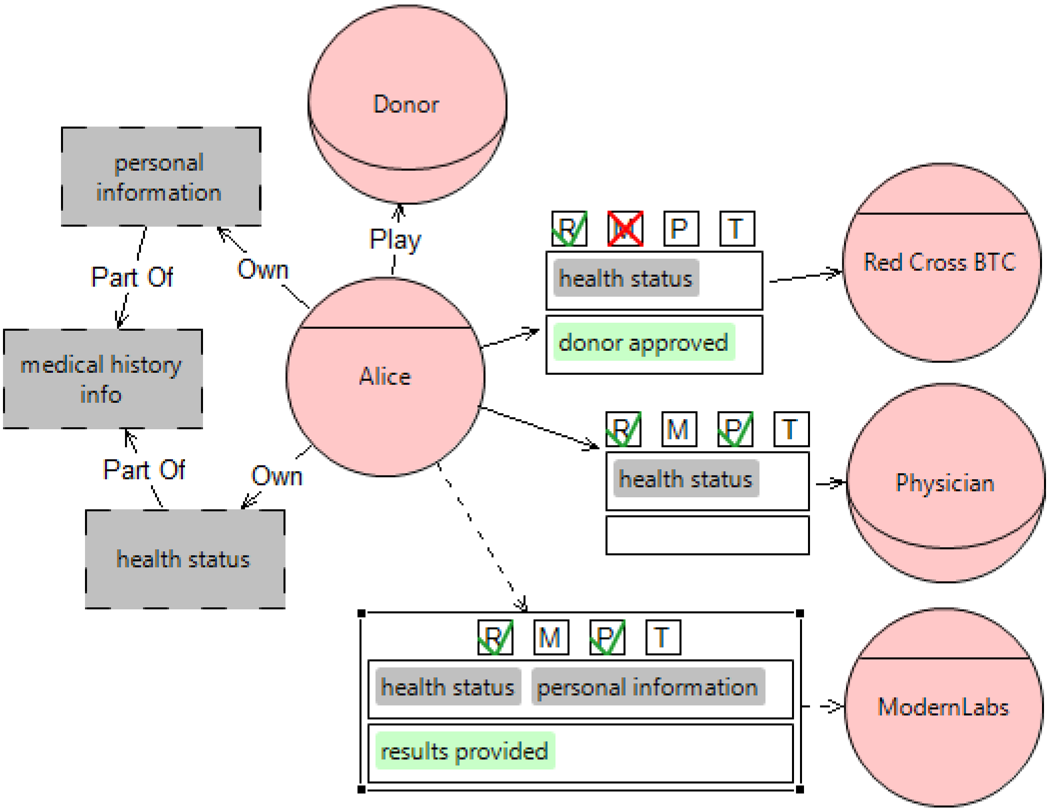


Figure 9. STS Authorization view Alice example

For social view example we will also take Alice example. Alice has one goal in scope of this scenario. She wants to donate blood. In order to achieve this goal Alice need to take tests and receive positive results. This means that we will decompose Alice's goal blood donated into goals test taken and positive results achieved using AND-decomposition. Also, to reach goal positive results received we need to read document test results. This is modelled by Read relation between corresponding goal and document. In scope of social view, it is also necessary to model that Alice plays role Donor. In order to take blood test Alice delegates her goal test taken to ModernLabs. This is modelled by goal delegation with no-repudiation, no-redelegation, redundancy and authorization rules to model Alice's security needs. ModernLabs in its turn transmits document test results back to Alice with confidentiality and integrity requirements to model security concerns on this document. After receiving test results Alice can transmit this document further to Red Cross BTC to donate blood. She does it with integrity, confidentiality and availability requirements. Model for this case presented below.

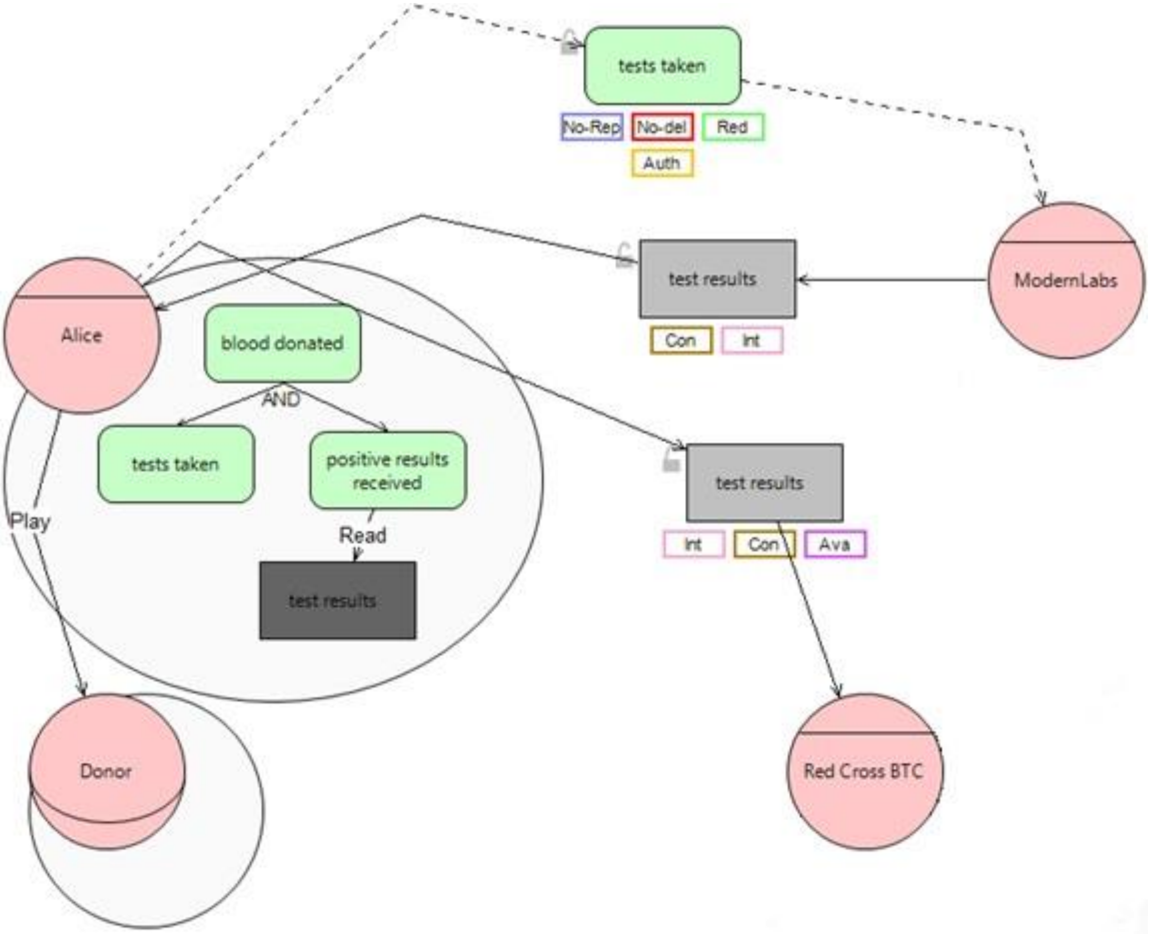


Figure 10. STS Social View Alice example1

3.2.4 Mapping to ISSRM

This chapter will clarify mapping between STS and ArchiMate based on definitions and meaning between concepts from both of them. Asset defined in ISSRM as *anything that is valuable and plays role in accomplishing organizations objectives* [9]. IS asset and Business assets are different types of assets, and main difference between them is that Business assets are immaterial and IS assets are material (including software). These concepts are presented in STS by Goal, Document and Information, since actors in STS see these concepts as valuable and since they are playing important role in achieving actor's goals. Security criterion in ISSRM *characterizes a security need* [9]. From STS side we set up security criterion over document transmit and goal delegation, therefore all types of this relationships will fall down to this bucket (except confidentiality, integrity and availability). Security objectives created using security criteria on business assets and typically describe security objective to be achieved [9]. Also, they are defined in terms of confidentiality, integrity and availability. From this points we can conclude that document transmit and goal delegation relationships which describe confidentiality, integrity and availability maps into this concept. Security requirement is defined as a *condition on the phenomena of the environment that we wish to make true* [9]. In STS we define this conditions through authorization element, which grants access to asset with several security constraints. Risk treatment decision, as defined in [9] – *decision to treat identified risk*. We don't have these concepts in STS. Control – *design means to improve the security by implementing security requirements* [9]. We don't have concepts for this in STS.

Table 2. STS mapping to ISSRM

ISSRM concepts		STS concepts
Asset-related concepts	Asset	Goal, Document, Information
	Business asset	Information, Goal
	IS asset	Document
	Security criterion	Document transmit related relationships (Authentication) Goal delegation related relationships (Availability)
	Security objective	Document Transmit related relationships (Integrity of transmission, Availability, Confidentiality of transmission) Goal delegation related relationships (Availability)
Risk-related concepts	Risk	-
	Impact	-
	Event	Event
	Threat	Threaten relationship
	Vulnerability	-
	Threat agent	-
	Attack method	-
Risk-treatment related concepts	Risk treatment	-
	Security requirements	Authorization
	Control	-

As for risk-related concepts, we can easily map ISSRM event from STS to event concept in STS, since they share the same name. To prove this point, we will check event definition from ISSRM – *risk event is an aggregation of threat and one of the vulnerabilities* [9], which is in other words stands for some harming event that threatens assets, and this is exactly what STS event is. Same mapping could be done for ISSRM threat concept. Threat concept defined as *an incident initiated by threat agent using an attack method to target one or more IS assets by exploiting their vulnerabilities* [9]. This perfectly maps on what threatens relationship in STS does. All other risk-related concepts are missing from STS. This is mainly because dominant aim of STS-ml is to capture and represent security requirements expressed by stakeholders [1]. This scope doesn't include much from risk related concepts. However, we still can model these concepts with some tricks. For example, we can introduce additional role of attacker (thus modelling threat agent) and reuse regular goals from STS to model vulnerabilities. Nevertheless, STS doesn't have dedicated elements to map for this ISSRM concepts.

3.3 What is ArchiMate

ArchiMate is an independent modelling language which provides instruments to analyse, describe and visualise relationships between various business domains. It offers one language to represent and combine IT systems, organisation structure, business processes, technical infrastructure and information flows.

ArchiMate could be used to model enterprise risk management (ERM) along with relationships and concepts of security. This standard is widely accepted and it provides enterprise architects with modelling constructs to logically connect business and technical architectures [4].

Core concepts of ArchiMate define three main types of entities, which represent real world entities. These entity types are:

- Active structure elements (this type of entities has behaviour)
- Behaviour elements (activities performed by Active structure elements)
- Passive structure elements (behaviour is performed on them)

Behaviour elements (services) usually represent functionality which system provide to its environment. Active structure elements (interfaces) usually represent endpoints through which system services to their environment. ArchiMate core concepts illustrated on figure 8. On this figure verb applies to closest entity. For example, Interface composes Active structure element, and Active structure element composed of Interfaces.

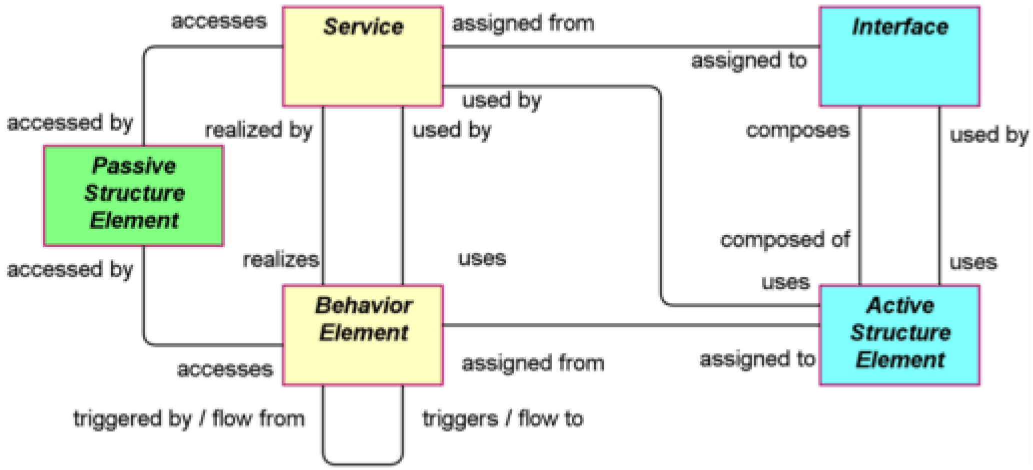


Figure 11. ArchiMate Core concepts [5]

ArchiMate generalises three core types of relationships between entities in the following groups:

- Structural relationships (model structural consistency between concepts. This group includes association, composition, access, aggregation, used by, assignment and realisation)
- Dynamic relationships (model dependencies between behaviour entities)
- Other relationships (neither structural, nor dynamic. Includes grouping, junction, specialisation)
- Based on speciality of each of them, ArchiMate defines three core layers, which are listed below:
- Business Layer (models business processes performed by business actors)
- Application layer (supports business layer with services realized by software)
- Technology layer (supports application layer with infrastructure services realised by hardware and system software)

Based on defined layers and concepts ArchiMate results in a nine cell framework illustrated on Figure 8

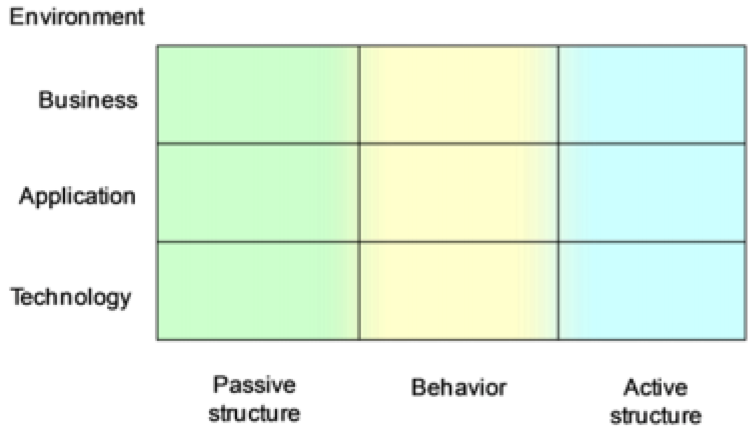


Figure 12. ArchiMate core framework [5]

In this paper we will not look into full ArchiMate framework, instead we will focus on Risk and Security Overlay (RSO) as it fits into scope of this work.

Risk and Security Overlay (RSO) to ArchiMate presented in [4] extends basic ArchiMate to model risk and security concepts with no additional notations [6]. Overall RSO consist of eleven new concepts represented by existing notations. ArchiMate Risk and Security Overlay general concepts (extracted from [4]) are visualised on figure 9.

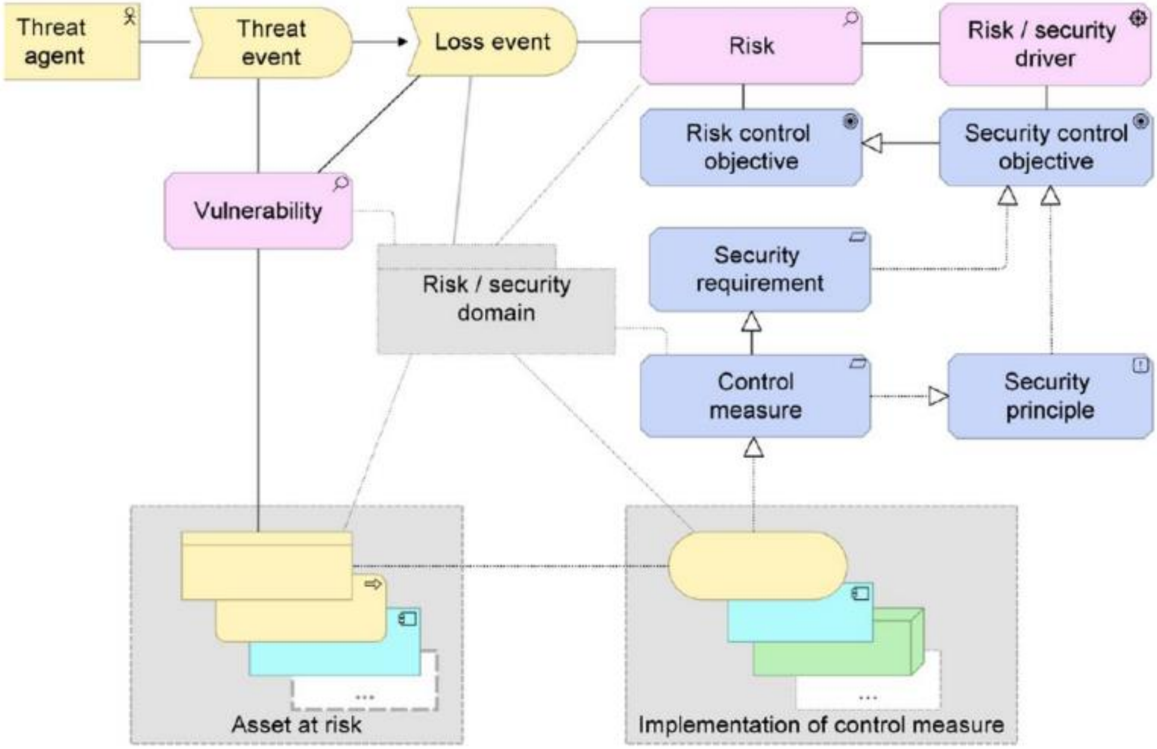


Figure 13. Risk and Security overlay for ArchiMate [4]

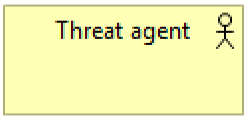
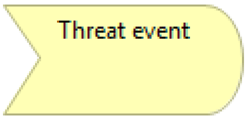
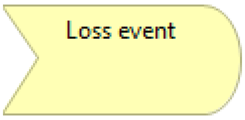
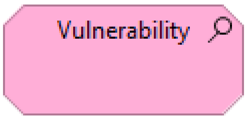
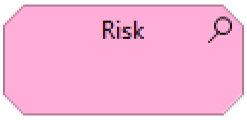
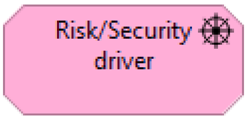
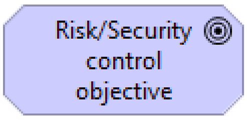
New concepts presented by RSO are following [6]:

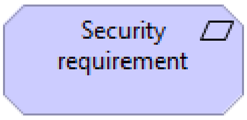
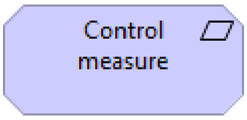
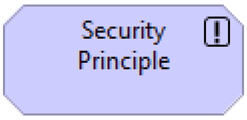
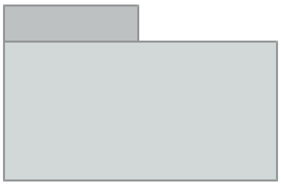
- Threat agent. Usually modelled by active structure elements such as business actor, business role, application component etc.
- Threat event (event that can influence asset)/Loss event (event that causes damage to asset). This one modelled by any business construct.
- Vulnerability. Modelled as assessment specialisation, because it is result of analysing weaknesses in architecture.
- Risk. Modelled by assessment construct.
- Risk/security driver. Modelled by driver construct and represents criteria to analyse risk.
- Risk control objective / Security control objective. Modelled by objective construct. Designed from risks and risk/security drivers.
- Security requirement/control measure. Modelled by requirement construct (control measure)
- Security principle (security policy). Modelled by principle construct.

3.3.1 Semantics and Concrete Syntax

Following table presents semantics and concrete syntax for STS. It makes perfect sense to combine semantics and concrete syntax in one table because we show how element looks like and explains meaning behind it in one place, which makes semantics and concrete syntax easier to remember and understand.

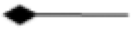
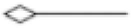
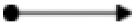
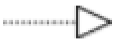
Table 3. ArchiMate RSO concepts

RSO Concept	**Core concept**	**Graphical notation**	**Explanation**
Threat agent	Any active structure element, e.g. business actor		Literally anything that can bypass security and cause harm. Can be intentional (attacker) or unintentional (executing wrong command).
Threat event	Business event		Event which can cause some effect on asset. Attack - specific type of threat events with attacker as threat agent.
Loss event	Business event		Any event that led to asset damage or loss.
Vulnerability	Assessment		Weakness which allows attacker to threat asset. Also represents probability that asset wouldn't be able to resist to threat agent's actions.
Risk	Assessment		Probable frequency and amount of future loss which is result of an action and/or inaction, foreseen or unforeseen [11].
Risk/security driver	Driver		Represents criteria by which risks are analysed.
Risk/Security control objective	Goal (3.0) /Objective (2.1)		Control objectives represents a high-level statement of intent with aim to mitigate risk. They are designed from risk and drivers and redesigned according to security requirement and principle.

Security requirement	Requirement		Abstract security requirement that must be met by system [7].
Control measure	Requirement		Same as security requirement but not that abstract, this one is more about implementation.
Security principle	Principle		Security qualitative statement that should be met by system.
Risk/Security domain	Group		Group of entities which share several characteristics relevant in risk management or security scope [6].

For modelling relations between this concepts core ArchiMate relations could be used. Semantic and concrete syntax for them presented in table below, derived from [7].

Table 4. ArchiMate relationship concepts

Concept	**Graphical notation**	**Explanation**
Structural relationships		
Composition		Represents that element consist of several other concepts.
Aggregation		Element groups several other concepts
Assignment		Allocates responsibility, execution or behaviour.
Realization		Represents that element plays important role in operation, creation, achievement of other more abstract entity.

Dependency relationships		
Serving	——>	Element provide its functionality to other element
Access	┈┈┈┈ ┈┈┈> <┈┈┈>	Represents ability of one element to perform actions on another passive structure element
Influence	- - +/- - ->	Represents the fact that element influence achievement or implementation of some motivational element.
Dynamic relationships		
Triggering	——▶	Casual or temporal relationship between elements.
Flow	- - - - -▶	Transfer from one element to another.
Other relationships		
Specialization	——▷	Models that element is exact kind of another element.
Association	——	For any unspecified relationship.
Junction	● (And) Junction ○ Or Junction	Serves to connect same type relationships.

3.3.2 Abstract syntax

In this chapter abstract syntax for specific concepts of ArchiMate Risk and Security Overlay and ArchiMate core relationship concepts will be discussed.

As making UML diagram for relationship concepts doesn't make any sense we will schematically represent them and briefly discuss.

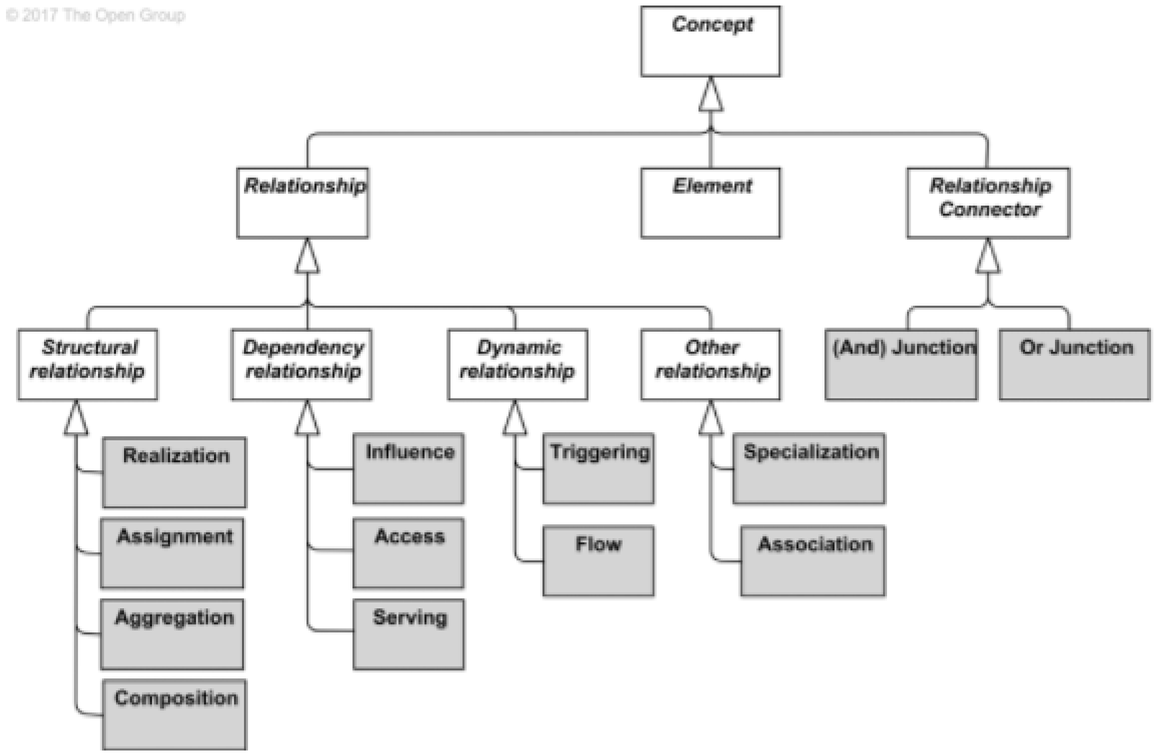


Figure 14. ArchiMate relationship concepts overview [7]

Each relationship should have exactly one "from" and one "to" elements. Following rules should apply for all relationships in ArchiMate:

- Relationship between two relationships are not allowed
- Relationships connected to relationship connector should be the same type
- Relationship connecting element with another relationship can be only aggregation, composition or association.
- Chain of relationships connected by relationship connector is valid only if a direct relationship between connected elements is valid.

Following diagrams will illustrate abstract syntax for ArchiMate Risk and Security Overlay. We split metamodel into two smaller ones to improve readability. First of them represent general concepts, while second illustrate risk related concepts in RSO. In first metamodel we introduce two new abstract concepts to catch behaviour - Element, which is abstraction over all elements in RSO, and Threat - which consist of Threat agent and Threat event. Second metamodel share the same approach with Element abstraction, which is abstraction over all other elements. Also, we introduce Requirement, Objective and Driver abstractions to sum up their corresponding elements.

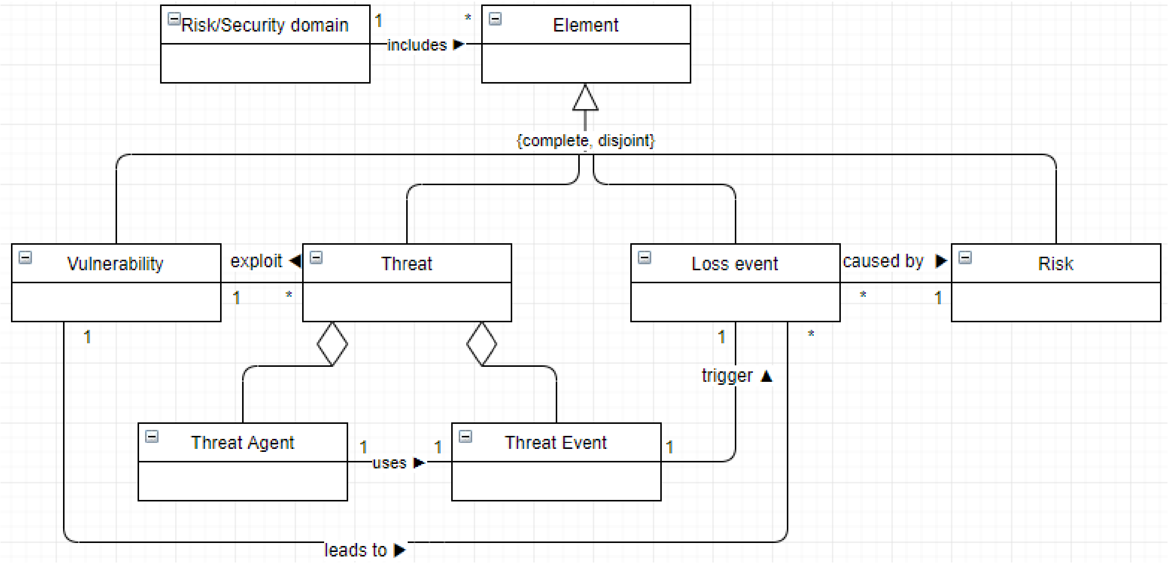


Figure 15. ArchiMate RSO metamodel

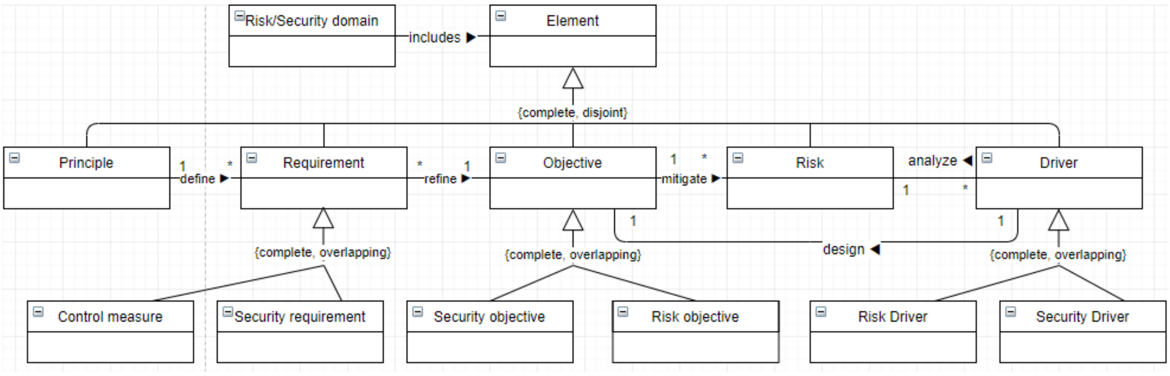


Figure 16. ArchiMate RSO risk related model

3.3.3 Application guidelines

This section will explain how to use ArchiMate in terms of ERM and modelling risk and security. As risk and security could be modelled with help of ArchiMate Risk and Security Overlay this chapter will provide basic example of usage on Healthcare scenario. We will not model whole healthcare scenario, but will do the same as we did in application guidelines for STS. We will model small part of scenario - Alice's test results in Modern Labs possession. Since ArchiMate approach focuses on threats and attacks, instead of information representation and possession as it is in STS, we can take models from STS application guidelines and think about threats with documents which tangibles Alice's information (Test results in our case). In the following example we will consider SQL injection to Modern Labs system as main threat for test results. Keeping this in mind we can easily identify threat agent - it will be attacker. With this context we can easily define threat event as attacker manages to break into Modern Labs system. We already set our main threat, therefore defining vulnerability is pretty straightforward - SQL injection is possible in Modern Labs system. Our asset at risk - test results. Keeping all this in mind we can set up our risk as all donor test results leaked. With stated risk we can define risk control objective - prevent information leak through SQL injection. Also we can define risk/security driver which will be the following: Donor test results should be secure and private. With drivers and risk control objective we can proceed to security control objective, which is: Modern Labs want to keep test results secure and private. From this point we can define Security principle and

Security requirement. Security requirement - Do not allow unauthorized access to data, Security principle -Test results should be secure and safe. And our last step - define Control measure, which should be practical solution to modelled risk. In our case control measure will be the following: Validate all inputs to prevent SQL injections. We haven't shown implementation of control measure in this example since this is out of scope of RSO and heavily involves core ArchiMate concepts.

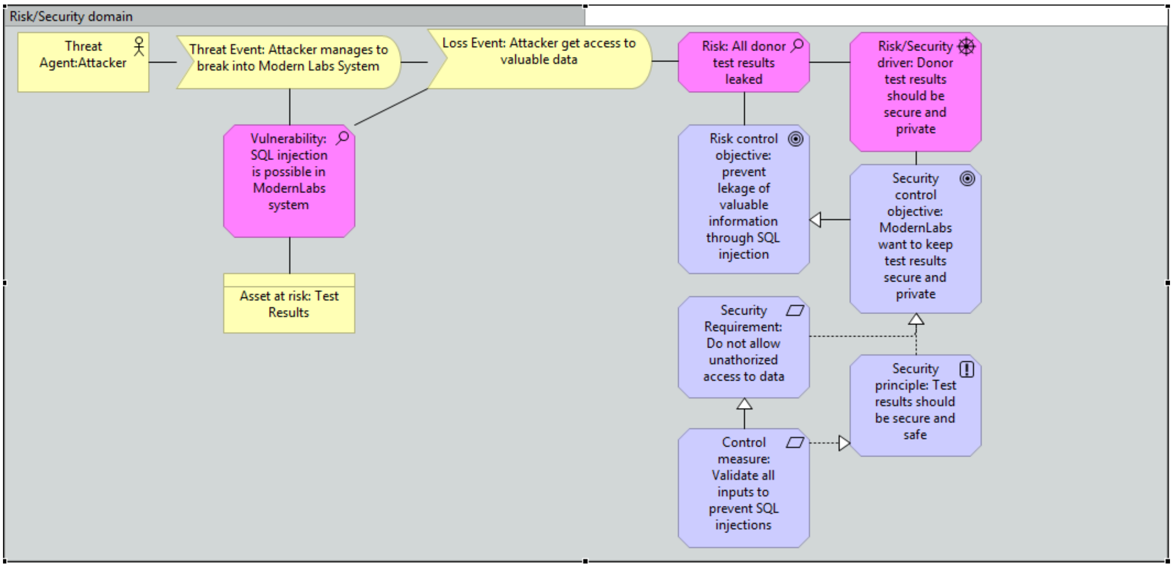

Risk/Security domain
Threat Agent:Attacker
Threat Event: Attacker manages to break into Modern Labs System
Loss Event: Attacker get access to valuable data
Risk: All donor test results leaked
Risk/Security driver: Donor test results should be secure and private
Vulnerability: SQL injection is possible in ModernLabs system
Risk control objective: prevent lekage of valuable information through SQL injection
Security control objective: ModernLabs want to keep test results secure and private
Asset at risk: Test Results
Security Requirement: Do not allow unathorized access to data
Security principle: Test results should be secure and safe
Control measure: Validate all inputs to prevent SQL injections

Figure 17. ArchiMate RSO usage example[4]

3.3.4 Mapping to ISSRM

Following chapter will map ArchiMate RSO concepts into ISSRM framework. Mapping is done based on explanations provided in [9] for ISSRM side and [6][4] for ArchiMate RSO side. Mostly mapping is pretty straightforward since concepts represented in both sides are pretty close to each other and sometimes even share a same name. However, this mapping could be incomplete and contain mistakes, therefore for double checking it is preferable to check information from RSO whitepaper [4].

From the table below we can see that mapping is almost full except of Event concept in ISSRM. Since ISSRM presents Event as combination of threat and vulnerability we can say that modelling threat agent performing a threat event in RSO equivalent to modelling Event is RSO.

[4] This and following RSO models are build in Archi available at: https://www.archimatetool.com/download

Table 5. Mapping of ArchiMate RSO into ISSRM [6]

ISSRM concept		**RSO concept**	**Construct from ArchiMate**
Asset-related concepts	Asset	Asset at risk	Any core concept
	Business asset		
	IS asset		
	Security criterion	Risk/security driver	Driver
	Security objective	Security control objective	Goal(3.0)/Objective(2.1)
Risk-related concepts	Risk	Risk	Assessment
	Impact	Loss Event	Business event
	Event	-	-
	Threat	Threat	Driver
	Vulnerability	Vulnerability	Assessment
	Threat agent	Threat agent	Active structure elements (business actor)
	Attack method	Threat event	Business event
Risk-treatment related concepts	Risk treatment	Risk control objective	Goal
	Security requirements	Security requirement	Requirement
		Control measure	
	Control	Implementation of control measure	Any core concept
		Security principle	Principle

3.4 Summary

This chapter focuses on modelling languages for security and security risk management. Initially we briefly described security risk-aware Secure Tropos and Mal-activities for UML. We didn't pay much attention to them as our main focus remain in STS and ArchiMate RSO. After that we in details goes through both STS and RSO, covering their semantics, concrete syntaxes and abstract syntaxes. We also present brief application guidelines for both STS and RSO, to improve understanding usage of this languages. Application guidelines are followed by mappings from STS and RSO to ISSRM. We need these mappings for following comparison between them and their integration, as it will be done under ISSRM concepts.

4 Alignment of ArchiMate RSO and STS

This section will cover primary contribution of this work – approach merged from both STS and ArchiMate RSO along with how this merging was done and comparison of STS and ArchiMate RSO.

4.1 Comparison criteria

As it was mentioned earlier we will make comparison and integration of STS and ArchiMate RSO under ISSRM concepts. We will pay attention to mappings of compared approaches into ISSRM, what was done in Chapter 2. Thus, comparison criteria would be following – to compare mappings of STS and ArchiMate into ISSRM, taking ISSRM concepts one by one and discussing their corresponding elements in STS and RSO. Moreover, we should pay attention to possible overlaps between ideas and concepts represented by elements of both approaches. In addition, during this process we should think about how concepts from both merged approaches can be merged together.

4.2 Comparison

This chapter will describe comparison between described in chapter 2 STS and ArchiMate approaches. Comparison is done under ISSRM concepts and through comparing STS mapping to ISSRM and ArchiMate RSO mapping to ISSRM.

Through looking into STS mapping into ISSRM (table 2) and ArchiMate RSO mapping into ISSRM (table 5) it is clearly seen that STS mapping lacks a lot of things in risk-related and risk-treatment related concepts. This fact could be used as a basis for merging this approaches. Besides that, as it becomes clear after general overview of both approaches, ArchiMate is much more abstract, with a good focus on risk-treatment and risk-related concepts. Whereas STS is more concrete, covering security requirements stated by stakeholders and not paying much attention to risks or treatment. In other words, ArchiMate RSO is good at covering abstract part and STS is good in covering concrete part of modelling. By this statement we can draw a separation line between two approaches. One way to use this is following: we can take ArchiMate RSO as basis for newly merged approach and enhance it with new concepts from STS, which will cover concrete part of modelling. Obvious downside of this approach is that newly introduced STS concepts will overlap over core ArchiMate concepts (since ArchiMate RSO is only add-on for core ArchiMate and relies on it to model concrete things). Another way is to consider STS as basis for new approach, therefore start at concrete level of modelling and enhance it with abstract concepts and risk-related ideas from ArchiMate RSO. This way seems more promising as it is not followed by any obvious downsides and it is easier to start from lower level of abstraction and move up to a higher one. We will follow second way in our merging.

To make comparison straightforward and clear we will follow divide and conquer principle, in other words, we will go one by one for each ISSRM concept and discuss corresponding concepts from STS and ArchiMate and how we can use best parts of them in merged approach.

4.2.1 Asset-related concepts

Asset. Asset is anything that is valuable for organization to achieve its goals [9]. ArchiMate RSO can represent this concept as any of the core ArchiMate concepts, thus making it a bit abstract and general in ArchiMate terms. STS in its turn uses goal, document and information to represent assets, which is more detailed. Asset in ISSRM has two sub-types – business assets and IS assets.

Business asset. In short, business assets are sub-type of assets and usually immaterial [9]. ArchiMate RSO doesn't distinguish assets and their sub-types as they are separated in ISSRM. In STS we can "draw" a separation line, and for business assets bucket goes Information and Goals (since they are immaterial). Information and goals also called as primary assets in STS [1], since they model the most important things for stakeholders. Since we decided to use STS as a skeleton for merged approach and ArchiMate RSO doesn't propose specific element for this concept – we can use STS elements in merged approach for business asset concept.

IS asset. Same as business assets, but material (including software). In ArchiMate RSO mapping they are the same as business assets and regular assets. In STS mapping documents (supporting assets) maps to IS assets. For merging purposes, we will use proposed by STS component, because RSO doesn't propose component for this concept.

Security criterion. Security criterion is defined in ISSRM as something that characterizes a security need [9]. ArchiMate RSO maps this concept for Risk/Security driver, whereas STS model this concept as some types of relationships. Since these elements are not overlapping and don't cause any conflicts we can use both of them in merged approach in a way that STS element remaining the same and introducing Risk/Security driver as abstract element to represent criteria by which we can analyze risk.

Security objective. Typically, security objectives describe security goal to be achieved (often in terms of confidentiality integrity and availability). RSO represent this concept as Security control objective element, which represents high-level statement of intent to mitigate risk, whilst STS model this concept in very explicit way – by setting confidentiality, integrity or availability security restrictions over different relations.

4.2.2 Risk-related concepts

Risk. As defined in ISSRM – risk is a combination of threat and vulnerability which cause harm to assets [9]. As we can see from STS-ISSRM mapping, STS doesn't have corresponding concept. ArchiMate RSO in its turn have exactly same concept which is also called risk. RSO definition for this concept is following - probable frequency and amount of future loss which is result of an action and/or inaction, foreseen or unforeseen [11]. Since this ISSRM concept is represented only in ArchiMate RSO we will use it in merged approach.

Impact. ISSRM definition for this concept is following – potential negative consequence of a risk that negates security criterion and harms assets when threat is accomplished [9]. As we can see in STS to ISSRM mapping – STS doesn't have representation of this concept. RSO represents this concept as loss event element, which is the same as impact in ISSRM. As we don't have any way to represent impact from STS we will use RSO way in our merged approach.

Event. ISSRM definition – aggregation of threat and one or more vulnerabilities [9]. This concept is completely missing from ArchiMate RSO as its whole model is built around one risk event (for each risk event separate RSO model), therefore making it irrelevant to represent on a model. On another hand, threat events are the only way to represent risk-related concepts in STS. This principal difference in two approaches could be crucial for our newly created merged approach, as it can be main point of touch for both STS and ArchiMate. In merged approach we can start by following STS guidelines, but when it will come to events and threats we can enhance STS way by hiding behind one STS threat event element huge variety of RSO concepts to model and represent this threat much better than it was modelled and represented by only one event element in original STS.

Threat. ISSRM defines threat as an incident initiated by threat agent with attack method to target assets [9]. STS model this concept as threaten relationship between threat event and asset. RSO model this concept as threat element which is the same as ISSRM threat. Hence, we can keep both concepts in merged approach as they are not overlapping. Threaten relationship from STS can connect asset under risk with threat event scope which will be extended by RSO concepts.

Vulnerability. From [9] – vulnerability is characteristic of asset that exposes a weakness in terms of security. As we can see from STS-ISSRM mapping - vulnerability concept is absent from STS. ArchiMate model this concept as Vulnerability element, which is pretty straightforward. This means that we will use RSO concept in our merged approach.

Threat agent. ISSRM definition for threat agent – agent that has intention to harm assets [9]. RSO represent this concept in a form of threat agent element, which is completely the same. As we can see from STS mapping STS doesn't have dedicated element for this concept. Of course we can use some tricks and use regular actor and role elements to represent actor whose goal is to harm assets, but we will not pay attention to this tricks as STS is not designed to do these things. Since only RSO propose corresponding concept for threat agent, we will use it.

Attack method. Attack method definition from ISSRM – it describes a standard means by which a threat agent executes threat [9]. This concept is completely missing form STS, as we can see from STS mapping to ISSRM. ArchiMate RSO represents this concept as threat event, with the same meaning as this concept have in ISSRM. As we don't have corresponding concept from STS, we will use one proposed by RSO.

4.2.3 Risk-treatment related concepts

Risk treatment. Definition for this concept in ISSRM is following – decision to treat identified risk [9]. As we can see from STS mapping, this concept is absent from STS. In ArchiMate this concept represented by Risk control objective. Since ISSRM risk treatment concept is represented only in RSO – our merged approach should use only it.

Security requirements. This concept is defined in ISSRM as condition on the phenomena of the environment that we wish to make true [9]. From STS side we set up this conditions by authorization element. STS authorization sets up read modify transmit and update access rules for assets in scope. RSO maps this ISSRM concept into two elements – security requirement and control measure. Here security requirement is some abstract requirement to security that must be met by system. Control measure is somewhat similar to security requirement, but on lower level of abstraction. In other words, control measure is more about implementation, not about abstract requirement. Since use cases of mapped elements are not overlapping we can take all of them into merged approach, with RSO concepts covering abstract part of security requirements concept and STS authorization element covering read create transmit and update rules over asset in defined scope.

Control. In ISSRM control is design means to improve security by implementing security requirements [9]. From ISSRM-STS mapping we can see that this concept is completely missing from STS. ArchiMate RSO in its turn have two elements to represent this ISSRM concept. One is implementation of control measure (which is low-level representation of security requirements implementation). Another one is Security principle. Security principle is security qualitative statement that should be met by system. Since STS doesn't have this concept we can easily use corresponding RSO concept in merged approach.

4.3 Integration

This section will cover STS and ArchiMate RSO integration along with semantics, concrete and abstract syntax for integrated approach. Also, application guidelines presented here, and benefits discussed.

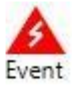
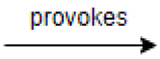
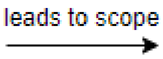
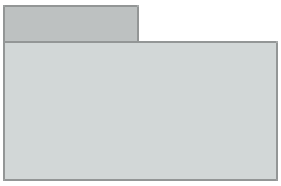
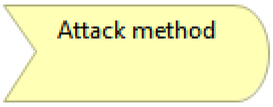
4.3.1 Discussion

As it was stated above, we will set up integration process as following: we will take STS as basic “skeleton” for integrated approach and enhance it with concepts and elements from ArchiMate RSO, following fundamentals of secure system modelling from ISSRM. Main reason for taking STS as basis is that STS lacks a lot of ISSRM concepts (especially risk-related), thus making it easier to fill gaps with concepts from RSO. Also, if we would take RSO as basis – we would face some overlaps between core ArchiMate parts (they are used by risk and security overlay) and STS parts. Another reason is that STS operates with more concrete terms than RSO, and from mine perspective it is easier to start with concrete things and advance to abstractions. Moreover, during comparison we noticed that Event concept from ISSRM could be used as integration point for STS and RSO. This could be used in the following way: we can keep STS flow as it is till it comes to events. Instead of representing threat as event (as it was in original STS) we can replace event element with RSO concepts. This would make perfect sense since whole RSO model are designed in scope of one threat event, and we are replacing threat event with RSO model. By doing this we will keep all the benefits of STS way in place, greatly improve threat modelling by RSO concepts (various RSO concepts make a huge difference comparing to only one threat event which was in original STS) and everything related to one threat event will be covered under separate RSO model, hidden behind this event, thus not adding mess to original STS models. Connection between event and threat event scope is settled through leads to scope relationship. In original STS one can use threat events only on social view (as STS focus is on social threats), but in integrated approach we propose to use threat events also in information view. This would allow us to model threat on information level, in addition to social level. Moreover, we propose to set up a new view level in STS and name it “risk and security view”. This view will cover all STS events which become exposed into RSO diagrams. In other words, this view will show separated by Threat event scope (new element) RSO models connected to their corresponding events (new element). On views with events there will be only events connected to their scopes (only scope without any inner elements), whilst on risk and security view each scope will be exposed and filled with RSO models. Also, we introduce provokes relationship from loss event to loss event to make integrated approach corresponding to ISSRM concepts.

4.3.2 Semantics and concrete syntax

Semantics and concrete syntax remains utterly same as STS and ArchiMate RSO semantics and concrete syntaxes. Main difference is that in case of integrated approach table representing semantics and concrete syntax for integrated approach will be a combination of tables for each approach. Thus, it is unnecessary to provide full abstract syntax of integrated approach. Instead we will provide a table semantics and concrete syntax of new or changed elements.

Table 6. Integrated approach semantics and concrete syntax (only difference)

Concept	**Graphical notation**	**Explanation**
Event (STS) => Event (Integrated)		Events as they were represented in STS is replaced by new concept. As STS events now become connection points between STS model and RSO model (or between Social/Information view and Risk and security view) we need to introduce new relationship between this element and threat event scope, to connect them. We will name this relationship “leads to”.
Provokes relationship		Provokes relationship is introduced to cover ISSRM’s provokes relationship in integrated approach.
Leads to scope relationship		Leads to scope relationship is introduced to connect events with their corresponding threat event scopes.
Asset at risk (RSO)	-	Got removed because become irrelevant in scope of integrated approach. Whole RSO model will be connected with asset at risk by threaten relationship from STS.
Risk/Security domain (RSO) => Threat event scope		Risk/Security domain element from RSO is redesigned and now covers up RSO model for each threat event and called threat event scope. Also, it gains new relationship “caused by” which is opposite of “leads to”.
Threat event (RSO) => Attack method (Integrated)		Since we already have Event element it would make sense to rename threat event element into something else, not to confuse people. For this purpose, we will use corresponding concept from ISSRM for which this RSO concept is mapped – attack method.

4.3.3 Abstract syntax

Abstract syntax for integrated approach remains almost the same as it was in original STS and ArchiMate RSO. All changes to abstract syntax are described in a table with concrete syntax and semantics. Connection point between STS and RSO models is settled up in a relationship between event and threat event scope.

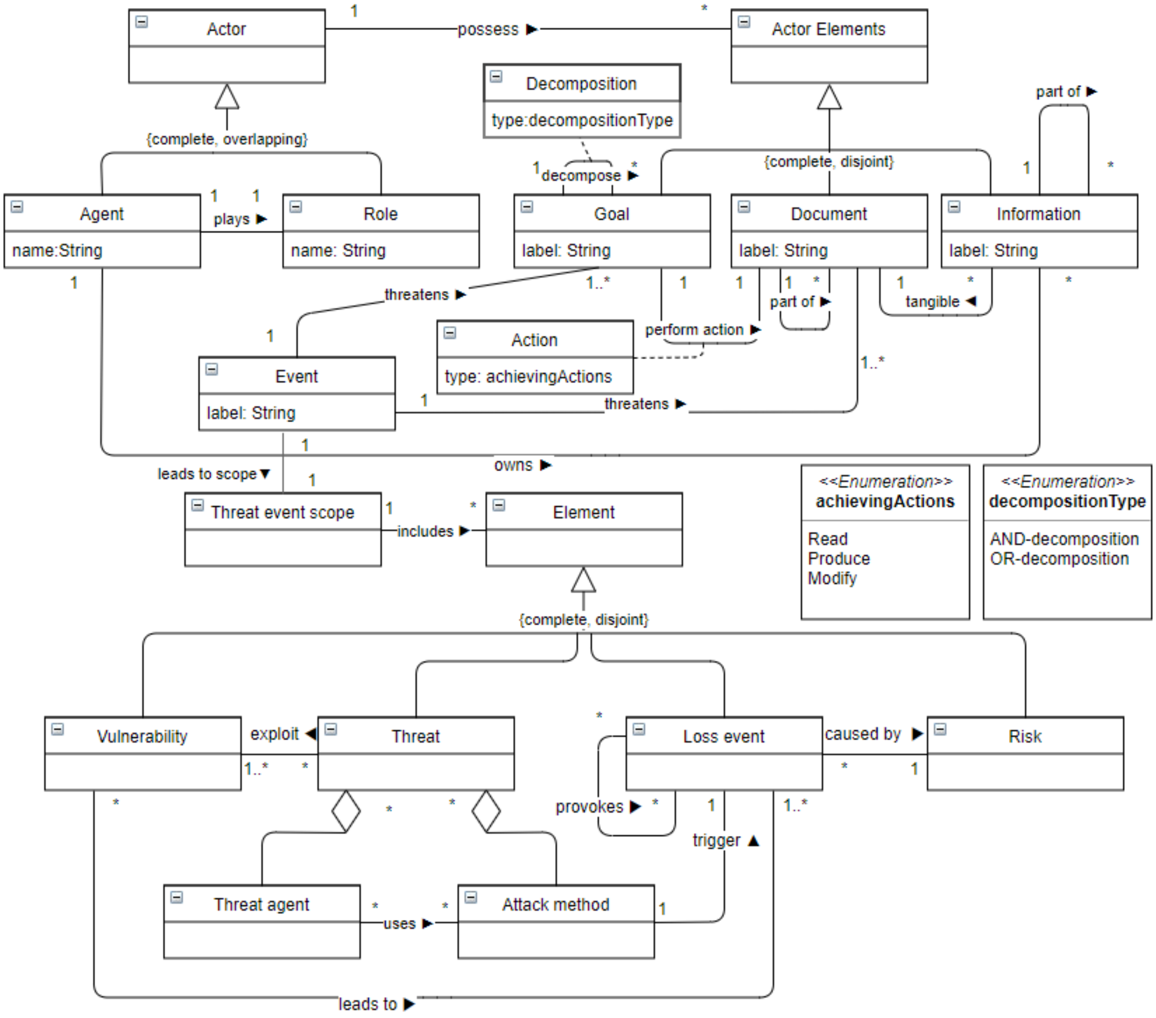


Figure 18. Integrated approach main metamodel

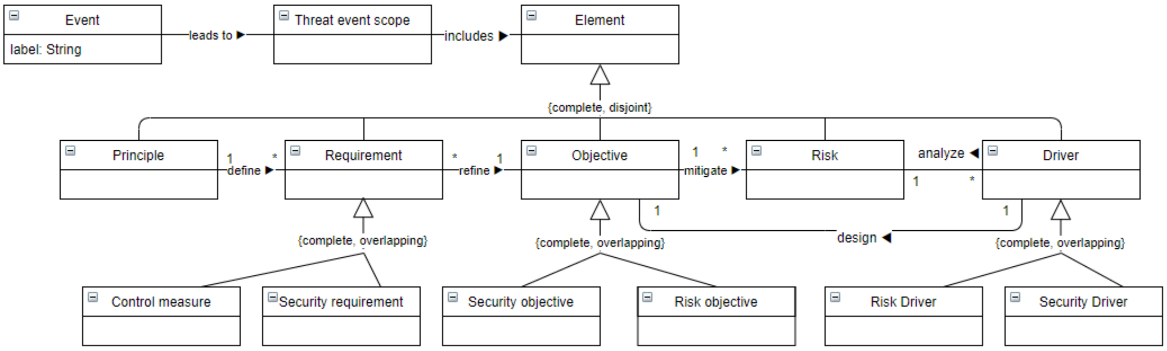


Figure 19. Integrated approach risk related metamodel

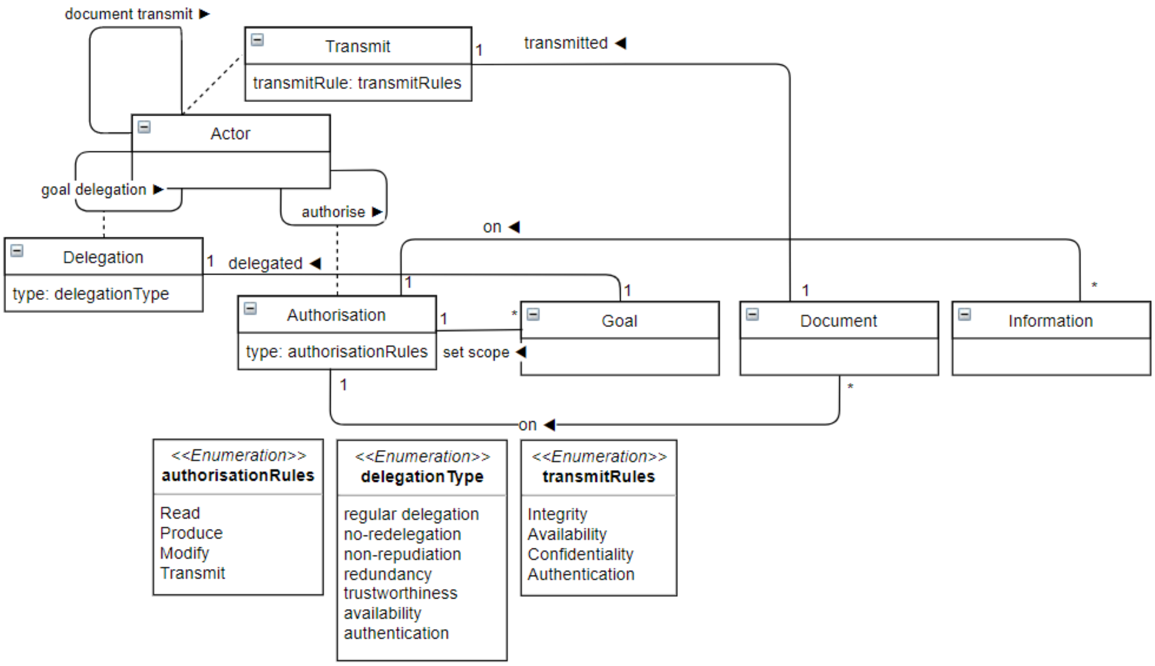


Figure 20. Integrated approach advanced elements metamodel

4.3.4 Application guidelines

This chapter illustrates usage example and gives brief application guidelines for integrated approach. Since integrated approach heavily depends on original STS and ArchiMate RSO, and reuse their mechanics – we will not duplicate descriptions of how to use STS and ArchiMate RSO approaches. Instead of that, we will show how STS and ArchiMate works together in merged approach, combining their benefits to enhance integrated approach.

For this example, we will take the same case which was used for application guidelines in STS application guidelines and ArchiMate application guidelines, and which is described in Background chapter. As we are building our integration from STS skeleton, following guidelines will be quite similar to those presented in STS application guidelines chapter. Main difference – new view, which is called "risk and security view". This view will assemble all identified threat events and presented them in a way of RSO models.

Information view follows the same guidelines as there are written for STS, but with some additions. As we propose to start using events on information view also, we will use one here. As we can see from figure below, they are used in exactly same way as they are used in classic STS. Here we have event "attacker breaks into system" which threatens document "test results". If there will be more documents in ModernLabs scope, then, obviously, event, which means breaking into system, would threaten all documents in one possession scope. But since we have only one document, this event threatens only one asset. To connect event on information view with corresponding RSO model on risk and security view, event and threat event scope should share same name.

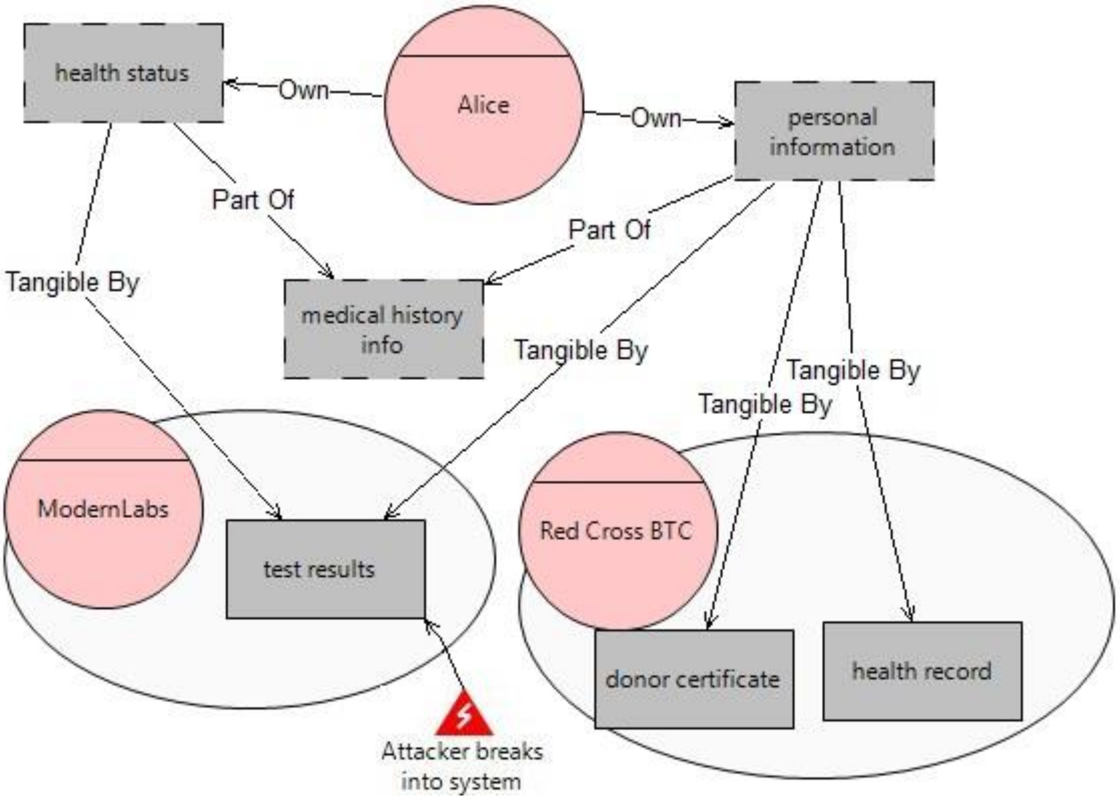


Figure 21. Integrated approach Information view

Authorization and social view remain the same as they were in STS, because we did not introduce any changes to this views in merged approach. Therefore, to check guidelines on how to build this views for our integrated approach we can easily check corresponding application guidelines for STS method in chapter 2.

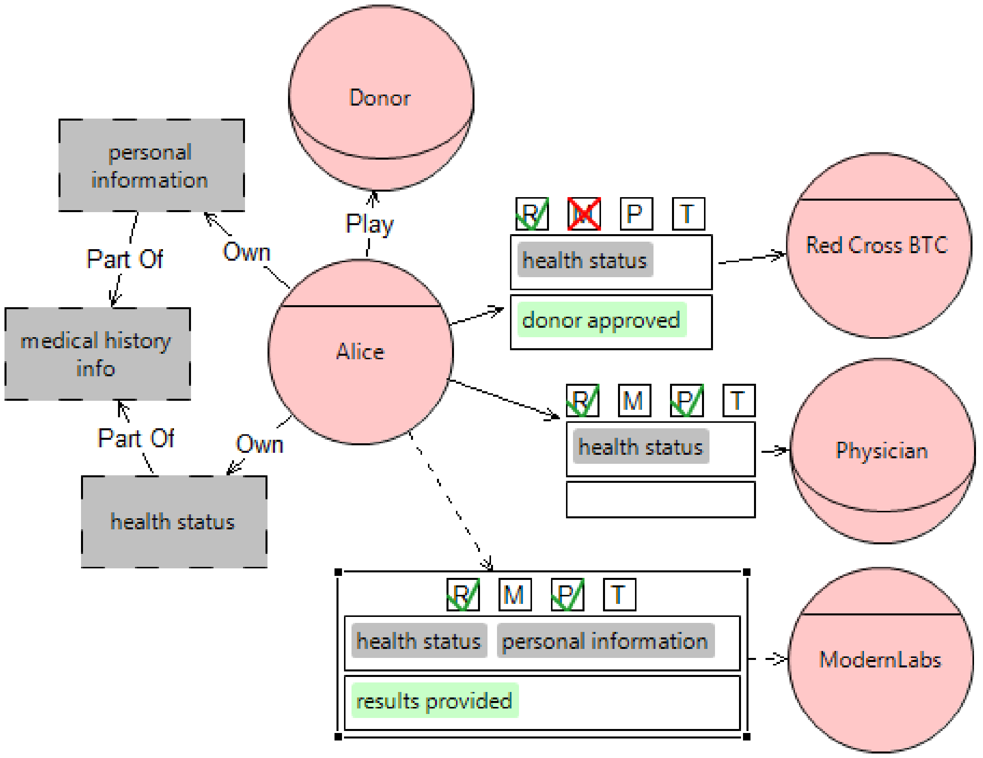


Figure 22. Integrated approach Authorization view

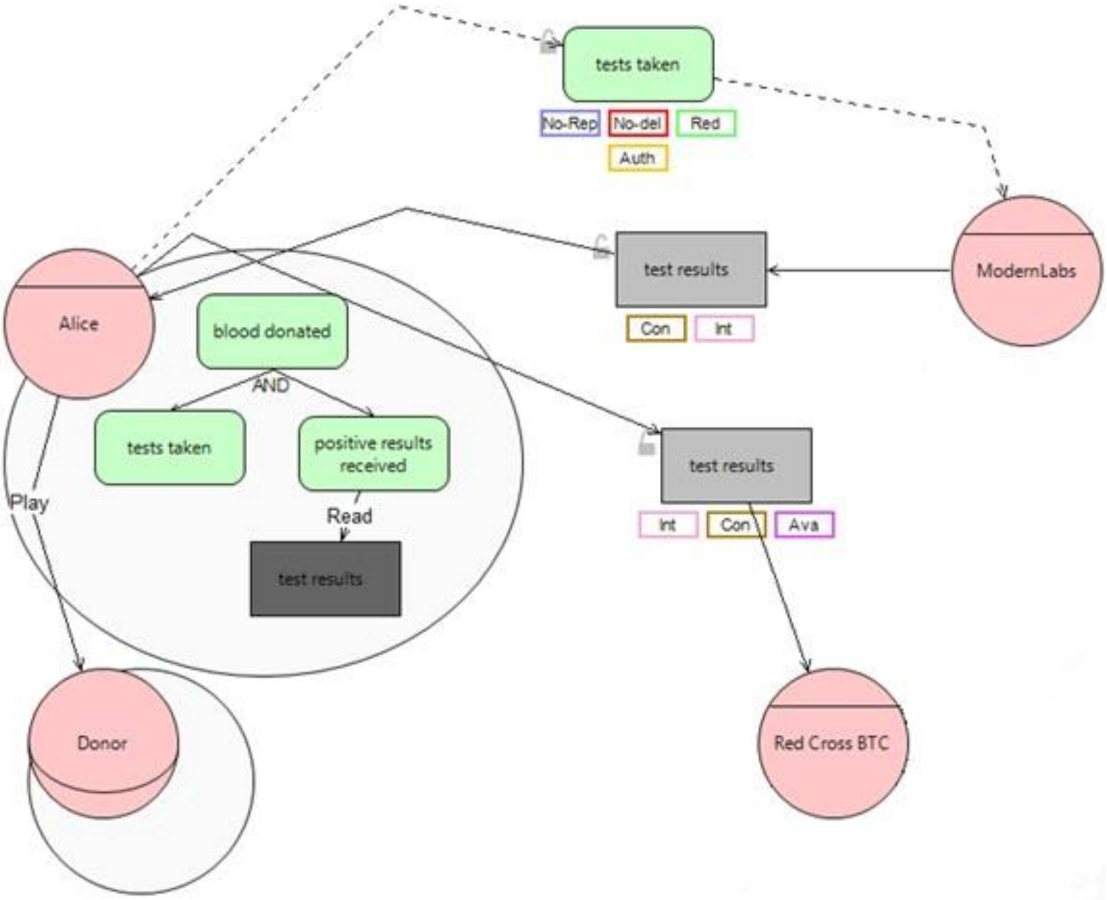


Figure 23. Integrated approach Social view

RSO models, assembled into integrated approach under risk and security view, remains quite same as they were in original RSO. They lost asset at risk concept as it becomes unnecessary, since every RSO model is connected with event on either information or social view and this event has a threaten relationship with asset at risk. In addition, we renamed original RSO threat event concept into attack method (to prevent misunderstanding between event and threat event concepts), keeping meaning behind this concept the same. Moreover, to correspond to ISSRM terms, we added possible provokes relationship from loss event to loss event (as some loss events could cause other loss events by itself). Other than that, building RSO model in integrated approach remains utterly same as it is in original RSO modelling. Guidelines for building original RSO models are described in application guidelines section for ArchiMate.

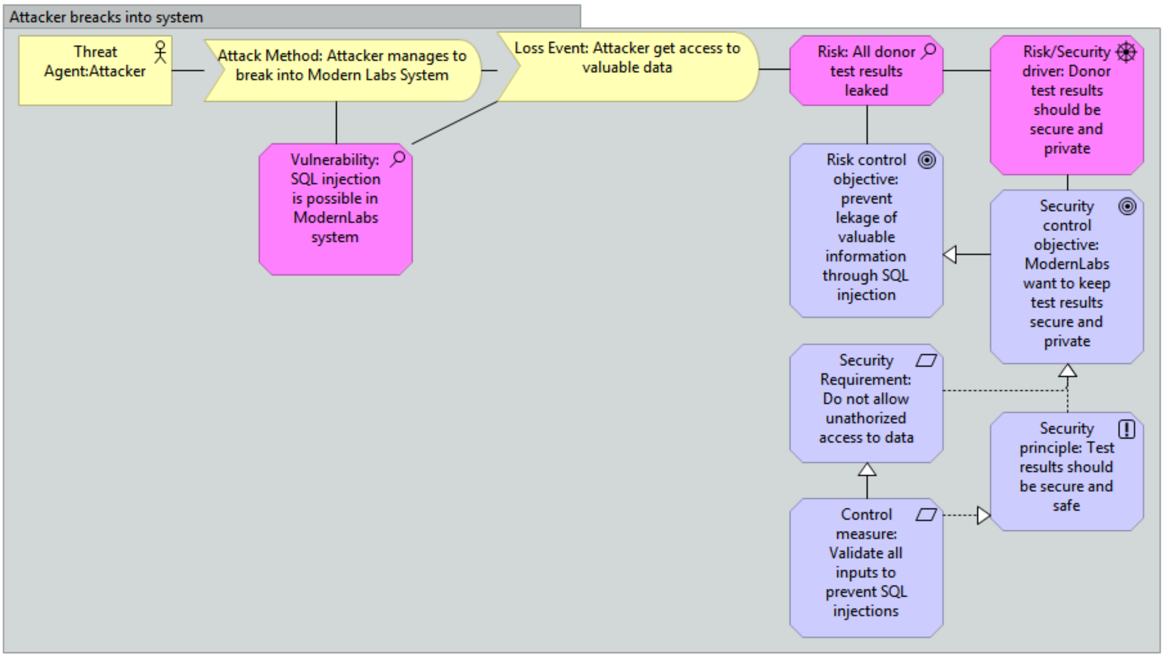

Attacker breacks into system
Threat Agent:Attacker
Attack Method: Attacker manages to break into Modern Labs System
Loss Event: Attacker get access to valuable data
Risk: All donor test results leaked
Risk/Security driver: Donor test results should be secure and private
Vulnerability: SQL injection is possible in ModernLabs system
Risk control objective: prevent lekage of valuable information through SQL injection
Security control objective: ModernLabs want to keep test results secure and private
Security Requirement: Do not allow unathorized access to data
Security principle: Test results should be secure and safe
Control measure: Validate all inputs to prevent SQL injections

Figure 24. Integrated approach risk and security view

4.3.5 Benefits

This section briefly describes benefits of integrated approach. As our integrated approach doesn't change or modify crucial points in STS or ArchiMate RSO and combine them both in a way that RSO abstractly models risk and security part and STS models social interactions and system overall on a more concrete level compared to RSO– we can say that integrated approach gains and combines all the benefits and advantages which STS and ArchiMate RSO had separately. Since our integration is, in fact, a combination of STS and RSO, it would make sense to highlight what exactly gains STS and RSO separately from this combination. From STS side: it is easy noticeable from STS description provided in chapter 3.2, that STS lacks a lot of concepts in risk-related field. STS operates only with threat events pointing onto assets at risk. Surely, this is not enough tools to build a good risk and security model which will capture various concepts from this field. This is weak point is STS and we help it by including ArchiMate RSO models to capture and represent risk and security concepts in our integrated approach. With including RSO models into STS we create a problem that including so many new concepts into STS could make models messy and way too crowded with new elements. We solve this problem by introducing new STS view level which we called "risk and security view". This view will completely consist from RSO models, separated one from another by threat event scope, and this models could be mapped backwards to events on information or social view by threat event scope name, which must be identical to event name. From RSO side: risk and security overlay depends on core ArchiMate to model everything out from scope of risk and security management. In our integration we replace this core ArchiMate parts with STS and applied risk and security overlay to STS. Based on this we can say that from this integration RSO gains all the benefits which STS can provide. Most significant among them is that RSO gains socio-technical perspective on modelling, which is completely missing from original ArchiMate RSO. Moreover, as we are taking into consideration only risk and security overlay for ArchiMate and not whole core ArchiMate – RSO doesn't have any tools or concepts to model anything out of risk and security field, it relies on core ArchiMate to do this. As we are integrating STS and RSO strictly (without core ArchiMate) – STS concepts replace core ArchiMate concepts to model systems which can have risks, which are modelled by RSO.

4.4 Summary

In this chapter we set up comparison criteria for STS and RSO. Comparison criteria is strongly bounded to ISSRM, as we are comparing and integrating this approaches under ISSRM and is following: to compare mappings of STS and ArchiMate into ISSRM, taking ISSRM concepts one by one and discussing their corresponding elements in STS and RSO. Moreover, this chapter presents actual comparison following defined comparison criteria. Based on this comparison we produce actual integration of STS and ArchiMate RSO into new integrated approach. This integrated approach turned out to be combination of both STS and ArchiMate RSO, keeping original flows of modelling and separating concerns of this two approaches on different view levels. Connection point is settled up in event concept from original STS, as it now leads us to whole RSO model, instead of one event concept, as it was in original STS.

5 Comparison of security risk management approaches

Following chapter will show usage of different security risk management approaches and compare them in terms of completeness in ISSRM scope. This comparison would evaluate that our integrated approach could be handy and useful and deserves to live.

5.1 Case description

As ongoing case for this chapter we picked up example from BPMN course in Tartu University, as it was already well investigated by me during this course and it is pretty complex. Detailed textual case description could be found on university website[5]. For better understanding of Prescription fulfillment process we provided BPMs, which were created during that course. They are provided in appendixes I and II.

Through reading case description and examining BPMs we found out that main risks could be the following: Violator could steal/modify private data from pharmacy system by exploiting intercept transmit channel, Violator could hack drug untilization review (DUR) check for his own needs and violator could hack insurance check for his own needs.

5.2 ArchiMate RSO

As we are not taking into consideration full ArchiMate, but only risk and security overlay for ArchiMate, following case will show models relevant only to overlay, without using core ArchiMate concepts. As our identified risks are stated in the case description we will consider them as our main threats for this case, as for all other also. Usual flow for building RSO model is following: we identify our most important assets and find vulnerabilities which can cause harm to these assets. By this we found our risks. From this we can define risk/security driver. At this point we can identify control objectives, and through them we can state security requirement for system. Out of security requirement and security principle we set up control measures to mitigate identified risk and close identified vulnerability. Each risk has separate RSO model to describe it. For SQL injection we come up with validating inputs. For intercept transmit channel we set up control measures as usage of secure connection and data encryption. Control measure for bruteforce attack – implementation of two factor authentication.

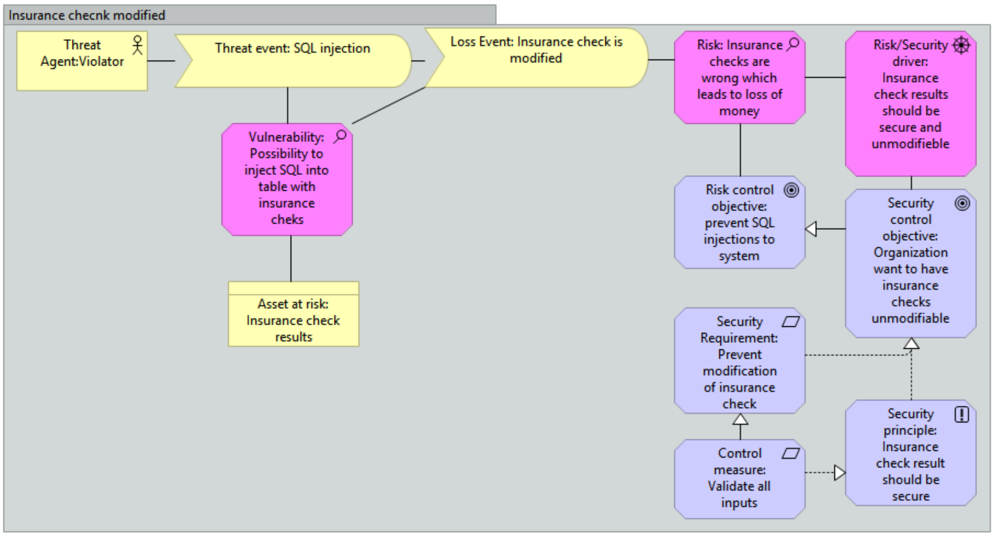

Insurance checnk modified
Threat Agent:Violator
Threat event: SQL injection
Loss Event: Insurance check is modified
Risk: Insurance checks are wrong which leads to loss of money
Risk/Security driver: Insurance check results should be secure and unmodifieble
Vulnerability: Possibility to inject SQL into table with insurance cheks
Asset at risk: Insurance check results
Risk control objective: prevent SQL injections to system
Security control objective: Organization want to have insurance checks unmodifiable
Security Requirement: Prevent modification of insurance check
Control measure: Validate all inputs
Security principle: Insurance check result should be secure

Figure 25. RSO Violator modifies insurance check results

[5] https://courses.cs.ut.ee/MTAT.03.231/2017_spring/uploads/Main/PrescriptionFulfillment.pdf

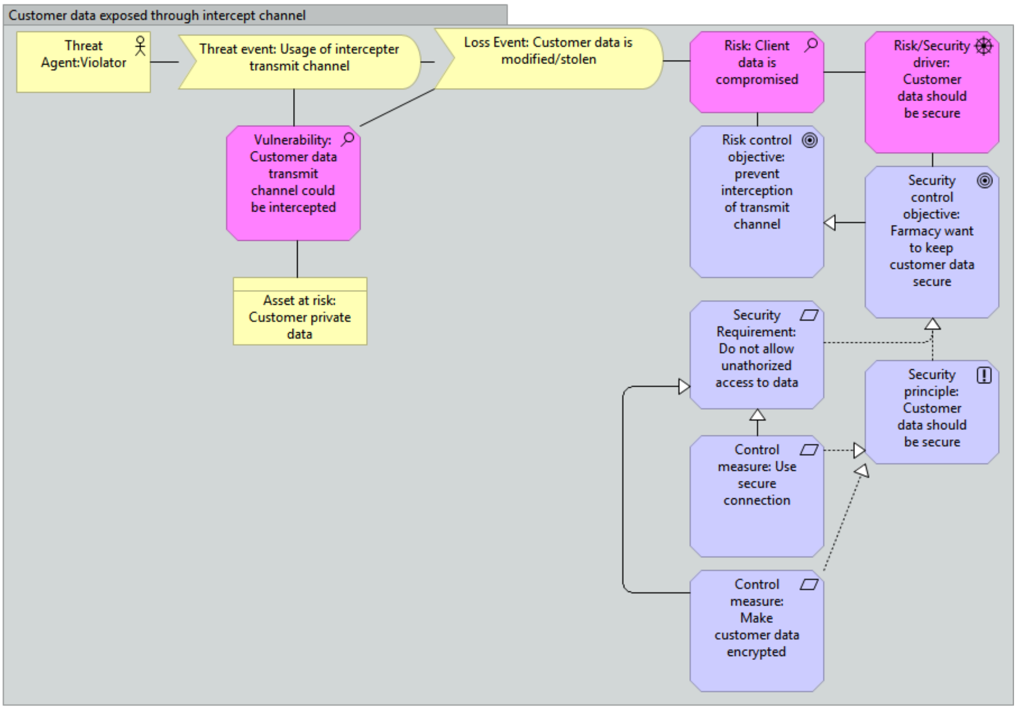

Customer data exposed through intercept channel
Threat Agent:Violator
Threat event: Usage of intercepter transmit channel
Loss Event: Customer data is modified/stolen
Risk: Client data is compromised
Risk/Security driver: Customer data should be secure
Vulnerability: Customer data transmit channel could be intercepted
Risk control objective: prevent interception of transmit channel
Security control objective: Farmacy want to keep customer data secure
Asset at risk: Customer private data
Security Requirement: Do not allow unathorized access to data
Security principle: Customer data should be secure
Control measure: Use secure connection
Control measure: Make customer data encrypted

Figure 26. RSO. Violator steals customer data

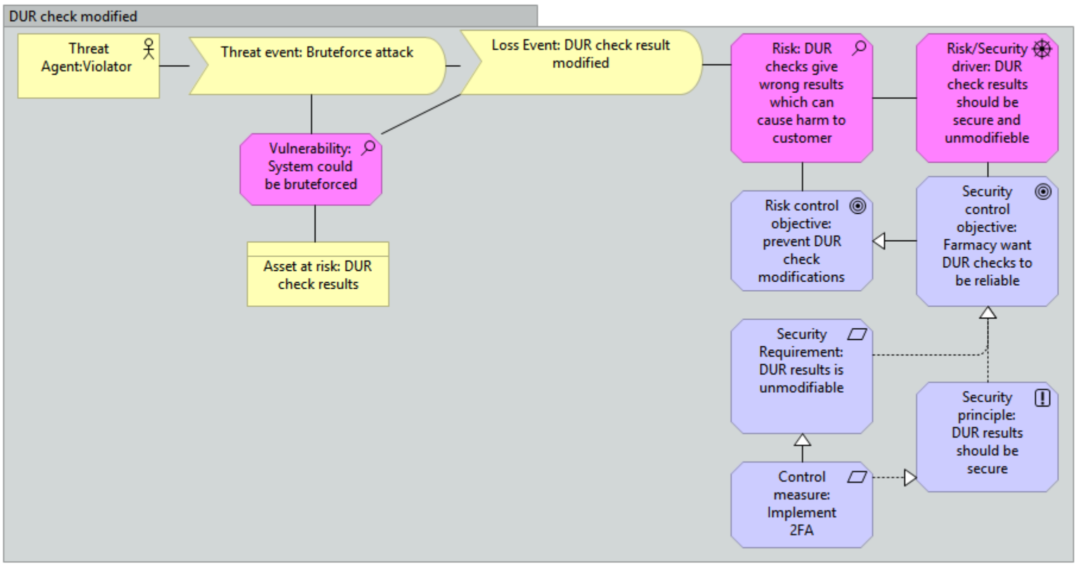

DUR check modified
Threat Agent:Violator
Threat event: Bruteforce attack
Loss Event: DUR check result modified
Risk: DUR checks give wrong results which can cause harm to customer
Risk/Security driver: DUR check results should be secure and unmodifieble
Vulnerability: System could be bruteforced
Risk control objective: prevent DUR check modifications
Security control objective: Farmacy want DUR checks to be reliable
Asset at risk: DUR check results
Security Requirement: DUR results is unmodifiable
Security principle: DUR results should be secure
Control measure: Implement 2FA

Figure 27. RSO. Violator modifies DUR check

5.3 STS

As we know from previous chapters STS have three different views to represent system. From information view we can see who owns what information and in which documents does this information become tangible. Through authorization view we can see which information is authorized to another role/agent along with rules settled over this authorization and in scope of which goal this is done. Most interesting view for us is social view, as only it can contain threat events. As we can see from it, Customer data in a form of prescription gets transmitted 4 times, where each transmission is threatened (intercept transmit channel). Also, we can clearly see where in the system DUR and insurance checks results are threatened, as well as information flows and security requirements settled over them.

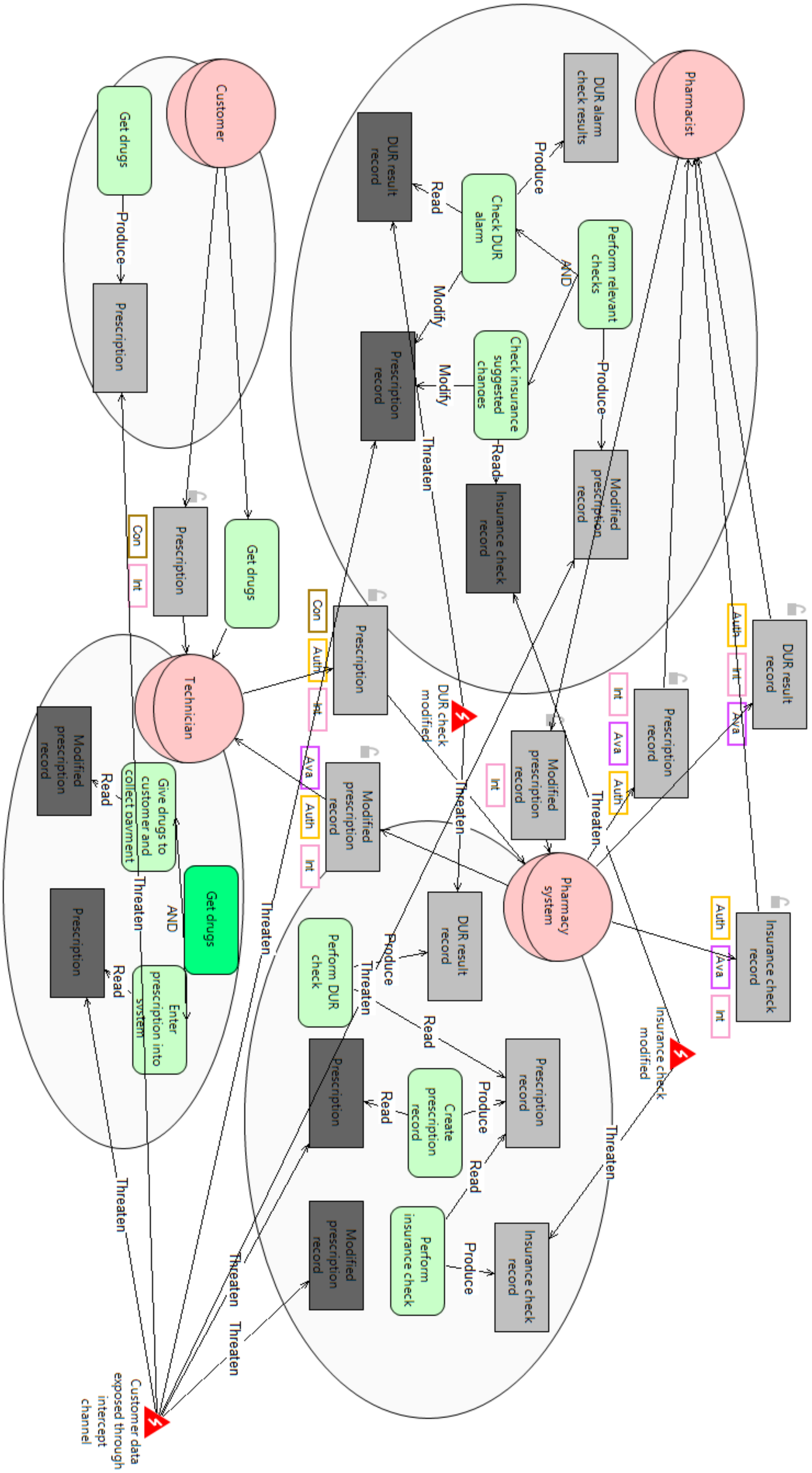


Figure 28. STS social view

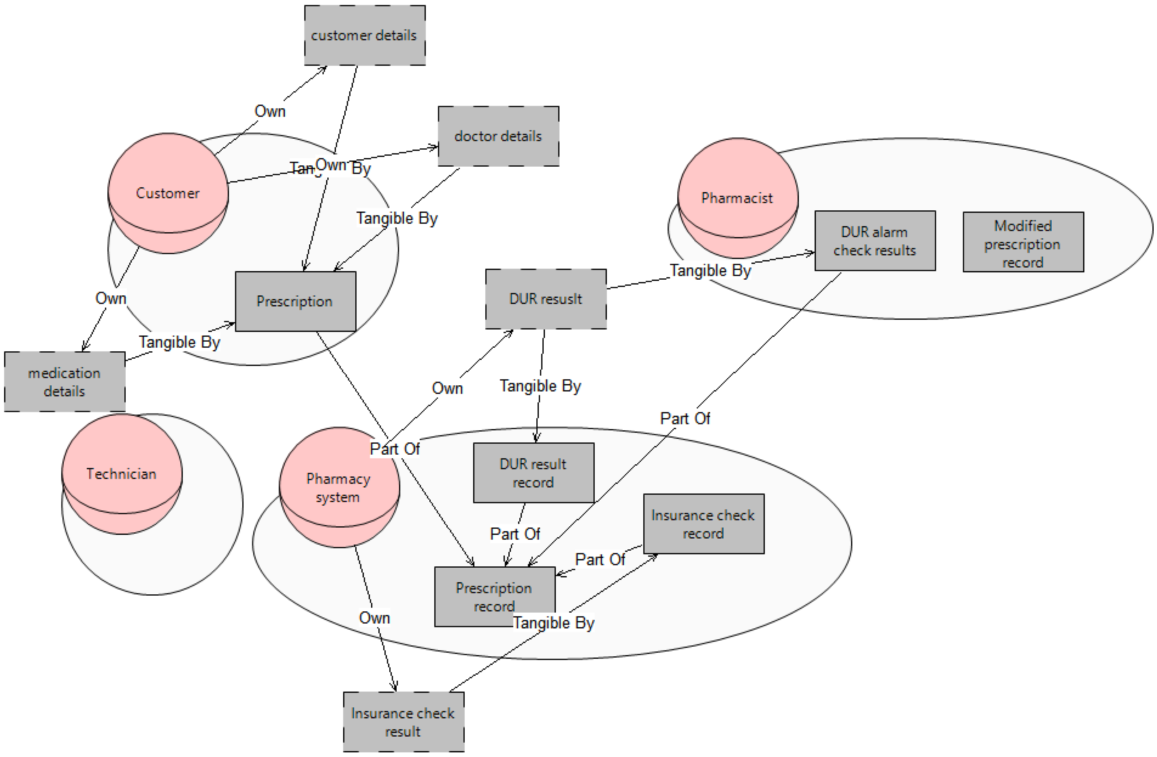


Figure 29. STS information view

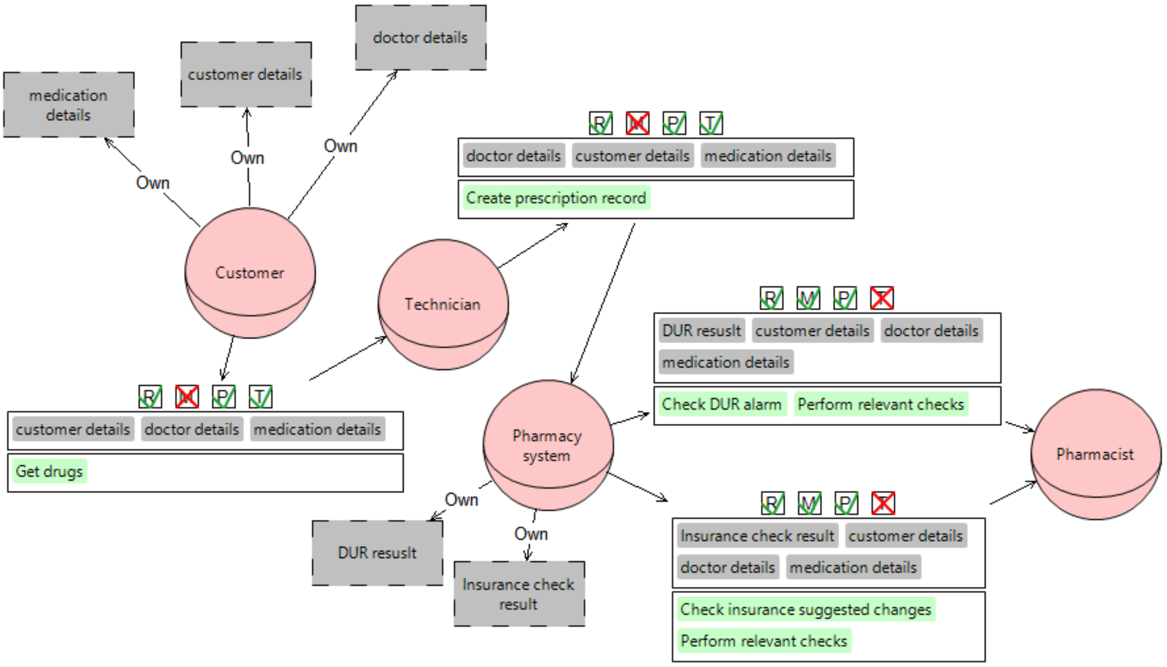


Figure 30. STS authorization view

5.4 Integrated

Overall, as integrated approach is a combination of STS and ArchiMate RSO, and reuse all their mechanics, nothing will change in STS views. Therefore, it is not necessary to duplicate models which are presented in STS subsection. Full model of integrated approach will be duplicated STS models plus models presented on figures in this subsection, where models presented in this subsection connected to threat events on STS models. Models presented in this subsection is our new risk and security view. This view is an expansion for classic STS events, which describe risks and vulnerabilities much better then only one event construct could. Also, this view introduces concept of risk treatment into STS, which were completely missing. However, models which represent new risk and security view doesn't differ much from RSO models. They don't have asset at risk element, and threat event scope name corresponds to threat event name on social/information view.

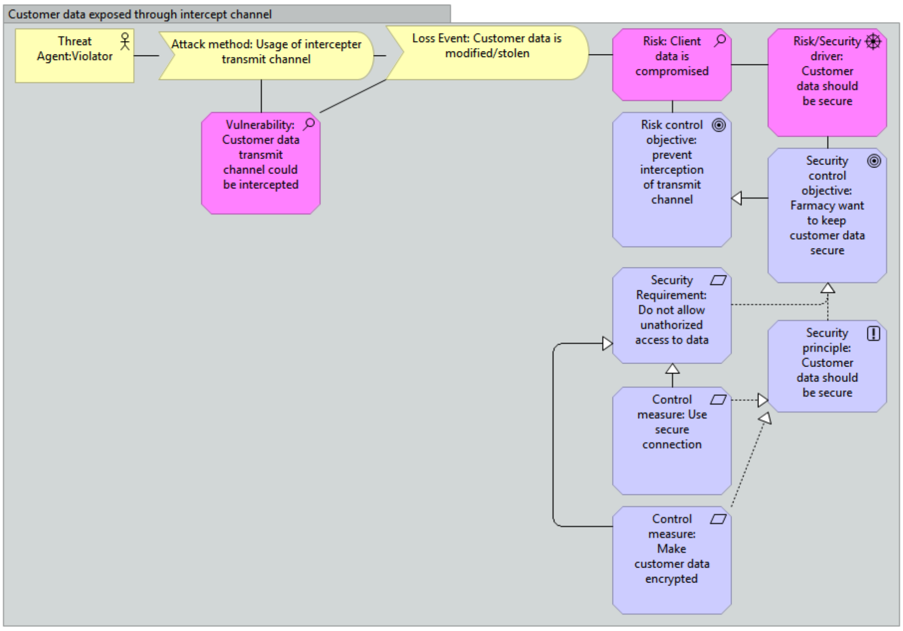

Customer data exposed through intercept channel
Threat Agent:Violator
Attack method: Usage of intercepter transmit channel
Loss Event: Customer data is modified/stolen
Risk: Client data is compromised
Risk/Security driver: Customer data should be secure
Vulnerability: Customer data transmit channel could be intercepted
Risk control objective: prevent interception of transmit channel
Security control objective: Farmacy want to keep customer data secure
Security Requirement: Do not allow unathorized access to data
Security principle: Customer data should be secure
Control measure: Use secure connection
Control measure: Make customer data encrypted

Figure 31. Integrated risk and security view. Intercept channel

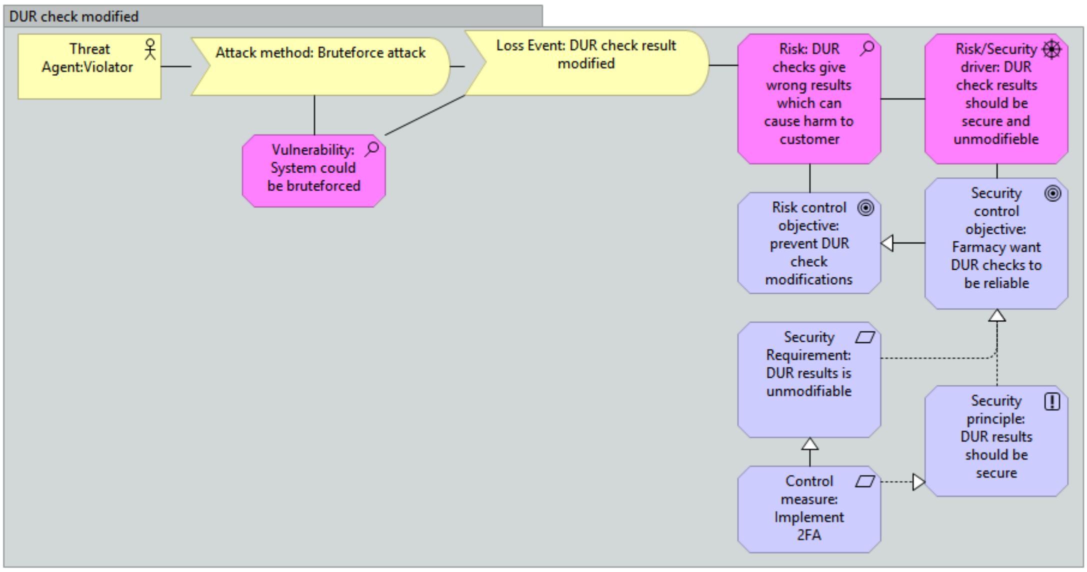

DUR check modified
Threat Agent:Violator
Attack method: Bruteforce attack
Loss Event: DUR check result modified
Risk: DUR checks give wrong results which can cause harm to customer
Risk/Security driver: DUR check results should be secure and unmodifieble
Vulnerability: System could be bruteforced
Risk control objective: prevent DUR check modifications
Security control objective: Farmacy want DUR checks to be reliable
Security Requirement: DUR results is unmodifiable
Security principle: DUR results should be secure
Control measure: Implement 2FA

Figure 32. Integrated risk and security view. DUR modified

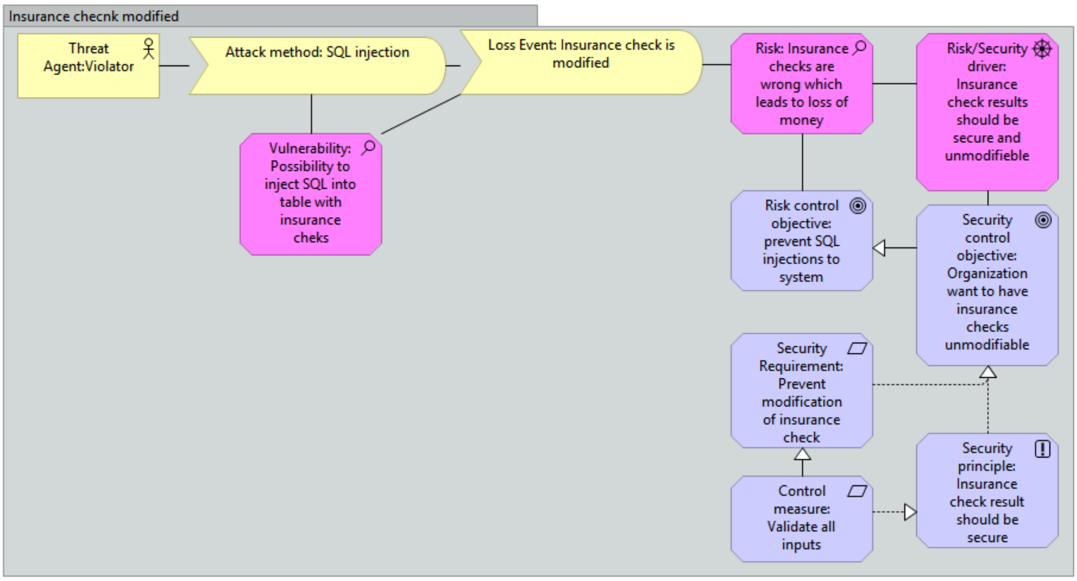

Insurance checnk modified
Threat Agent:Violator
Attack method: SQL injection
Loss Event: Insurance check is modified
Risk: Insurance checks are wrong which leads to loss of money
Risk/Security driver: Insurance check results should be secure and unmodifieble
Vulnerability: Possibility to inject SQL into table with insurance cheks
Risk control objective: prevent SQL injections to system
Security control objective: Organization want to have insurance checks unmodifiable
Security Requirement: Prevent modification of insurance check
Security principle: Insurance check result should be secure
Control measure: Validate all inputs

Figure 33. Integrated risk and security view. Insurance check modified

5.5 Security risk-aware Secure Tropos

Secure Tropos process consist of the following stages: assets identification, risk analysis, definition of security criterias. We will follow same steps to model our case here. Model below represents refined assets and security criterion for our case. Most important task on the model below is manage pharmacy storage, which is split into three lower level tasks. This model also presents security criterions “integrity of prescription” and “integrity of checks” settled up over the business assets. In Secure Tropos we can set up security objectives by softgoals and refine them using security constraints. Secure Tropos satisfies security constraints through achieving security goals, which are also system assets. In our example security constraints are “only if authorized” is satisfied by achieving secure goal “Access only for authorized users”. Plan “check login and password” satisfies security goal, therefore improving overall security of system.

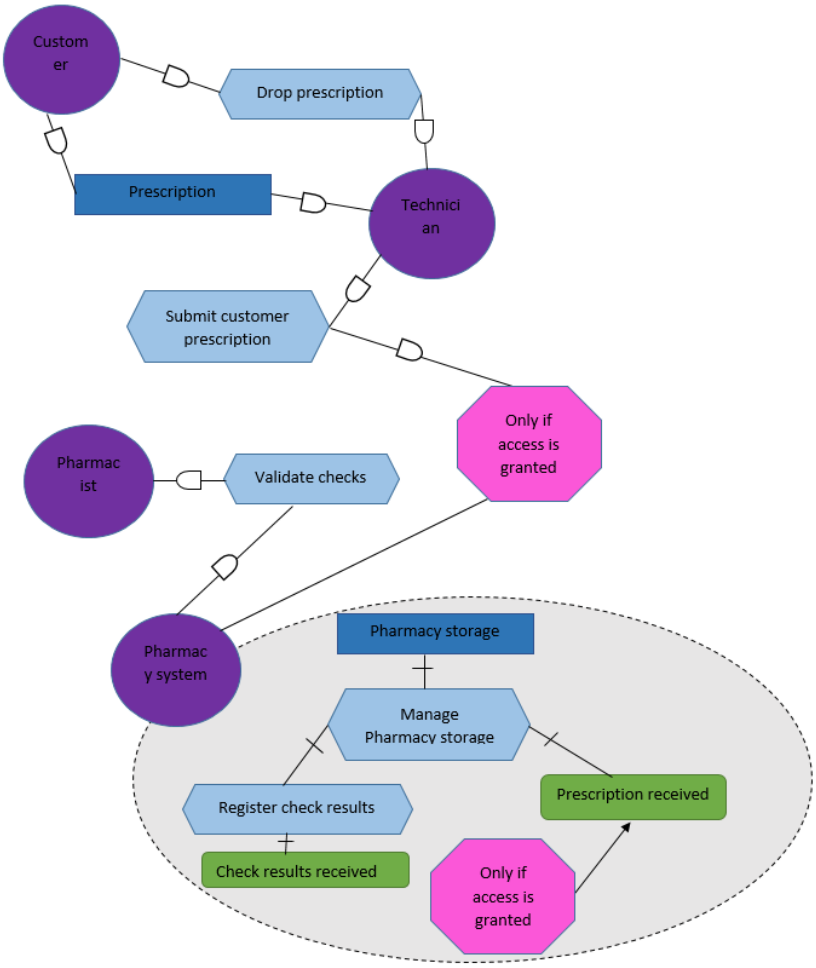


Figure 34. Business assets modelling for Secure Tropos

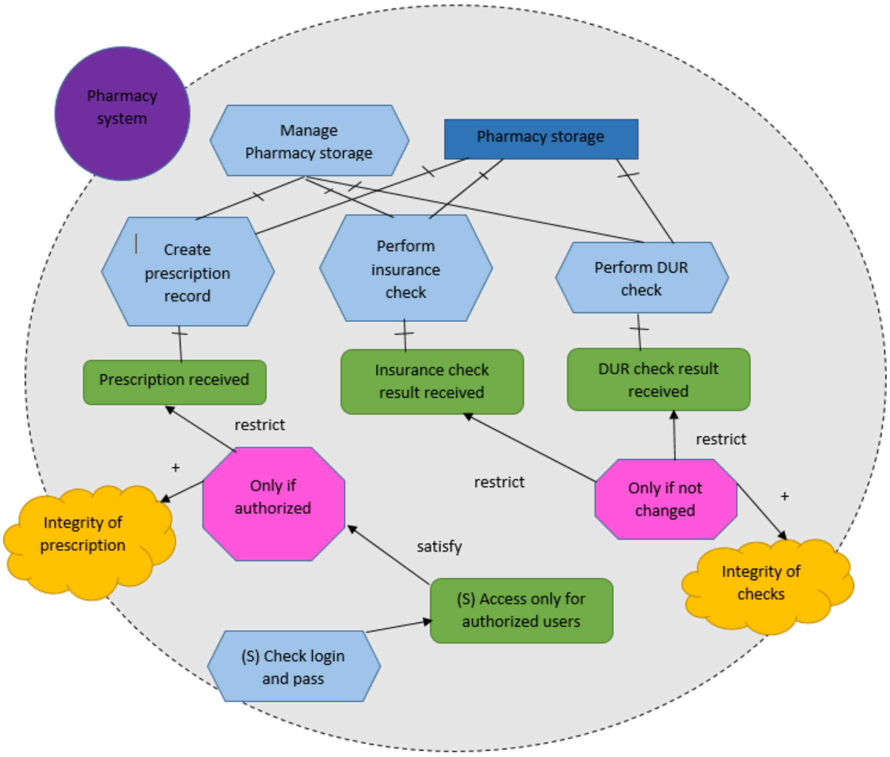


Figure 35. Assets identification for Secure Tropos

On the second step we include risk into our model. On figure below we illustrate security event “intercepted channel usage” which impacts “integrity of prescription”, “brute force attack” which impacts “integrity of DUR check” and “SQL injection” which impacts “integrity of insurance check”. Detailed description of this risks is provided on the figure with potential attack scenario. Within it violator uses threats (“collect info and steal customer data”, “collect info and change DUR result”, “insert SQL that change insurance check”) which attacks “pharmacy system storage”, which holds organization’s assets such as “prescription record”, “DUR result”, “Insurance check result”. Also, on this model we can see that violator’s attack method “check password repeatedly” exploits vulnerability in “check login and password”, attack method “repeatedly input SQL” exploits vulnerability in “store insurance check in DB”, attack method “catch prescription thorough intercepts channel” exploits vulnerability in “send info through channel”.

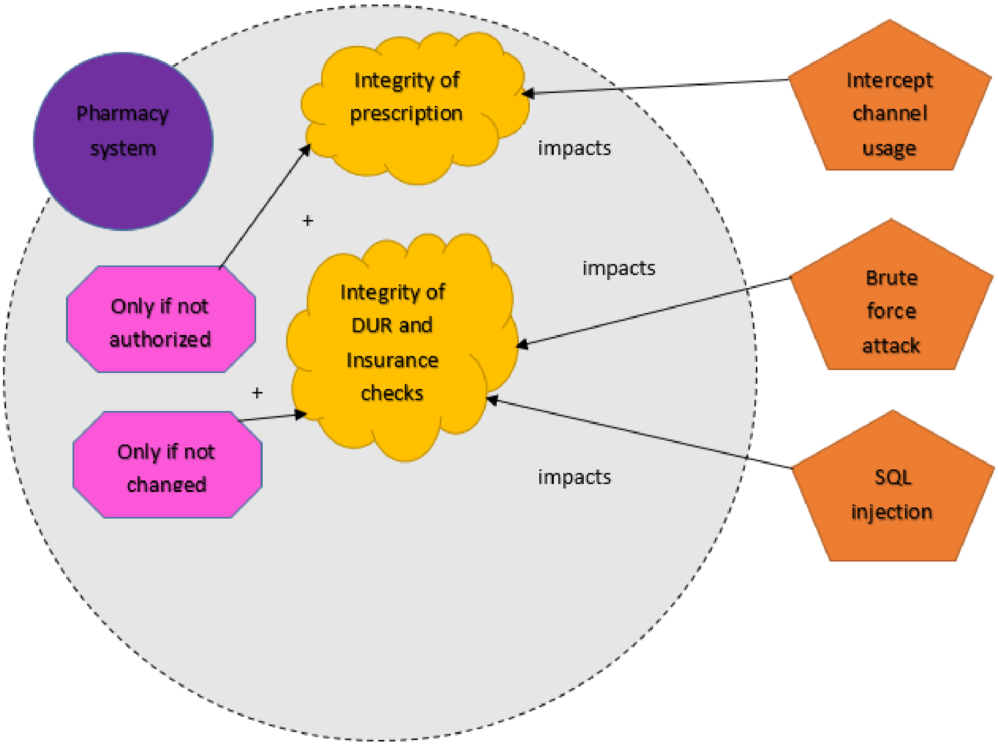


Figure 36. Identification of security risk for Secure Tropos

On this stage we should define security criterias. So, to mitigate identified risks we made risk reduction decisions. For example, to mitigate brute force attack we implement two factor authentication, to protect ourselves from SQL injections we will validate all inputs, to mitigate interception of channel, we will use secure connection.

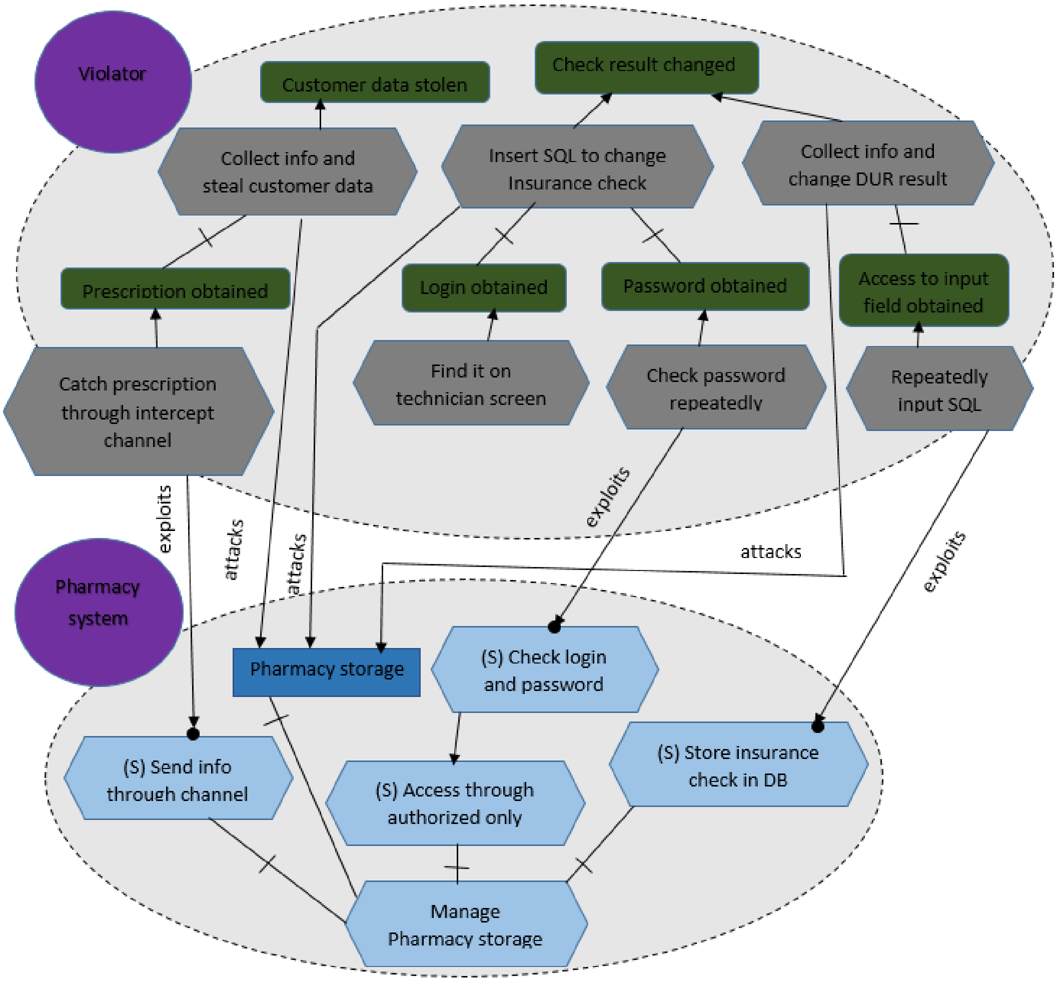

Violator
Customer data stolen
Check result changed
Collect info and steal customer data
Insert SQL to change Insurance check
Collect info and change DUR result
Prescription obtained
Login obtained
Password obtained
Access to input field obtained
Catch prescription through intercept channel
Find it on technician screen
Check password repeatedly
Repeatedly input SQL
exploits
attacks
attacks
exploits
attacks
exploits
Pharmacy system
Pharmacy storage
(S) Check login and password
(S) Send info through channel
(S) Access through authorized only
(S) Store insurance check in DB
Manage Pharmacy storage

Figure 37. Potential attack scenario for Secure Tropos

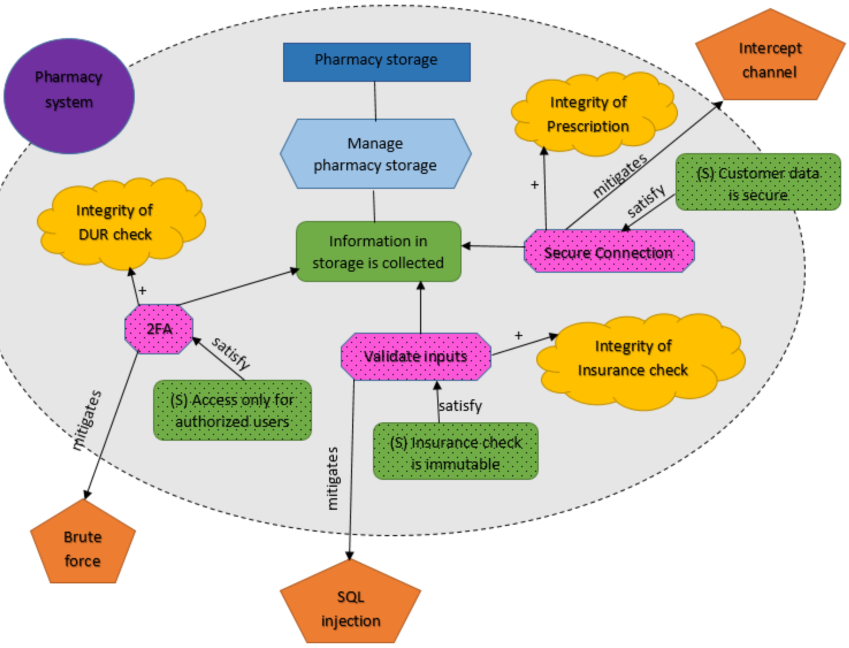


Figure 38. Security requirements definition for Secure Tropos

5.6 Design and Engineering methodology for Organizations (DEMO)

We have on it three composite actor roles which forms our environment and three elementary actor roles which forms our composition. As we can see from model, technician and pharmacist interact with pharmacy system and DUR and Insurance checkers in a form of transactions. These transactions usually transmit information through them. It is important to keep in mind that transaction should be connected by initiator link with actor role which initiates this transaction and by executor link with actor role that should perform this transaction. To define what information is transmitted we connect elementary role actors with production banks (they represent information). Through this connection it is possible to found out what information is transmitted. Also, if elementary or composite actor role relies on some information or produce it – we shout connect this role with production bank by access link. To model violator, we define brand new composite actor role named violator. This role is indicated to abuse system vulnerabilities and harm assets in various ways. We model violators attacks by defining transaction kinds with indicating in their names used vulnerability and harmed asset. These transactions are initiated by violator and executed by corresponding actor role from organization boundary (composition role). In addition, as violator gaining access to corresponding information, we should connect him with appropriate production bank. Unfortunately, DEMO does not provide any tools to highlight risk mitigation, therefore they are missing from this model.

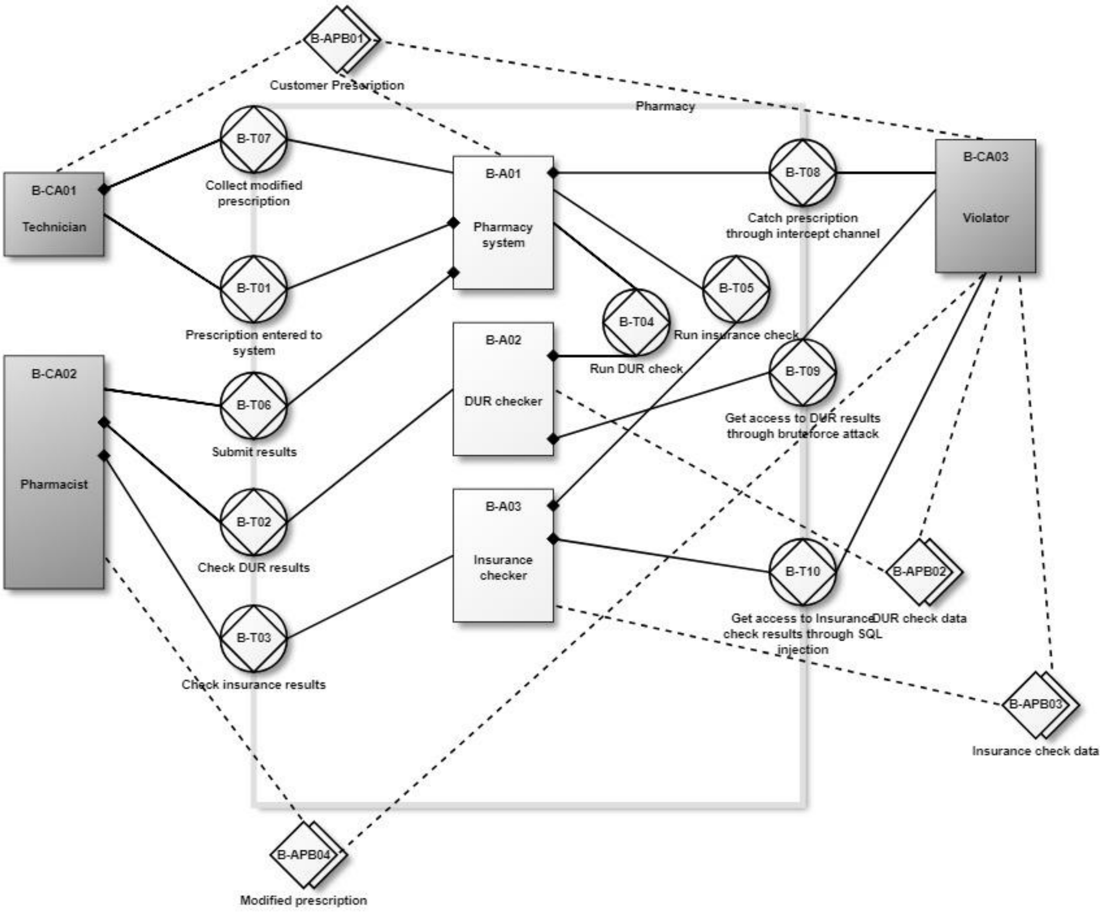


Figure 39. DEMO construction model[6]

[6] This model is built in online tool available at: http://www.modelworld.nl/

5.7 Comparison

Comparison criteria. To validate our integrated approach, we will compare all described in this chapter approaches to modelling in terms of *completeness* with respect to ISSRM. This means that we will go through ISSRM domain model and all possible relationships presented in this model and explain how are they presented in current approach. This would give us results of how complete described approach is in terms of ISSRM, and this results could be compared with each other. Through this comparison we will validate how complete our integrated approach is in terms of ISSRM.

We will distinguish ISSRM concepts by risk treatment related, risk related and asset related concepts. **Asset related concepts** include following elements: *Asset, Business Asset, IS Asset, Security criterion* and *Security objective*, along with following relationships: *constraint of* and *supports*. Overall 5 elements and 2 relationships. **Risk related concepts** include following elements: *Attack method, Threat agent, Threat, Risk, Impact, Event, Vulnerability,* and along with following relationships: *uses, exploits, leads to, targets, characteristic of, provokes, harms, negates* and *significance assessed by*. Overall 7 elements and 9 relationships. **Risk treatment related concepts** include following elements: *Risk treatment, Security requirements* and *Control* along with following relationships: *Refines, implements, decision to treat* and *mitigates*. Overall 3 elements and 4 relationships.

Note that in sections 5.7.1 – 5.7.5 we will highlight concepts from discussed approach with italic and ISSRM concepts will remain as plain text.

5.7.1 ArchiMate RSO

As for elements completeness we can check RSO mapping presented in section 3.3.4 to check what ISSRM elements are presented in RSO.

Asset related concepts. According to table 5 RSO contains 5 out of 5 ISSRM asset related elements. ISSRM supports relationship presented as *composition/aggregation* relationships between assets elements. Constraint of relationship is presented in a form of connected *risk/security driver* to asset. Thus, RSO fully represents ISSRM asset related concepts, 5 pout of 5 elements and 2 out of 2 relationships.

Risk related concepts. According to table 5 RSO contains 6 out of 7 ISSRM risk related elements. Because RSO doesn't have event element (*threat event* relates to attack method, not to event), it is obvious that RSO will not have leads to relationship presented. Uses relationships is modelled by connecting *threat agent* with *threat event*. Threat exploiting vulnerability relationship is visualised by connection between *threat event* and *vulnerability*. Characteristic of relationship is modelled by *association* relationship between *vulnerability* and *asset*. As threat is modelled as a combination *threat agent* and *threat event*, we can say that ISSRM targets relationship is represented by *association* between this combination and *asset* (through *vulnerability*). Harms relationship is represented by *loss event* connected with *asset* through *vulnerability* by *association* relationship. Negates relationship becomes tangible by connection between *loss event* and *risk/security* driver. Provokes relationship is missing from RSO as it is not intended to model impacts (*loss event*) to cause (provokes) other impacts. Significance assessed by relationship is captured in a form of connection between risk (RSO *risk*) and security criterion (*risk/security driver*). To summarise, RSO in scope of SSRM risk related concepts represents 6 out of 7 elements and 7 out of 9 relationships.

Risk treatment related concepts. According to table 5 RSO contains 3out of 3 ISSRM risk treatment related concepts. ISSRM implements relationship is captured by *realization* relationship between *control measure* and *security principle*. Mitigates relationship from ISSRM is represented in a form of a sequence of connections from *security requirement* to *risk* (goes through *security control objective* and *risk control objective* to reach *risk*). Refines relationship is captured by relationships from *control measure* and *security requirement* to *risk control objective* through *security control objective*. Moreover, *risk/security control objectives* are refined by *security requirement* and *principle* after they are initially designed, according to [6]. Decision to treat relationship is modelled by *association* relationship between *risk* and *risk control objective*. To sum up, RSO in scope of risk treatment related concepts represents 3 out of 3 elements and 4 out of 4 relationships.

Total. Overall, RSO covers 14 out of 15 elements and 13 out of 15 relationships.

5.7.2 STS

As for elements completeness we can check STS to ISSRM mapping in section 3.2.4 to see what elements from ISSRM are presented in STS.

Asset related concepts. Based on table 2 we have 5 out of 5 elements present. ISSRM support relationship presented in form of *tangible by* relationship. ISSRM constraint of relationship is missing from STS. To sum up, in scope of asset related concepts STS covers 5 out of 5 elements and 1 out of 2 relationships.

Risk related concepts. Based on table 2 we have 2 out of 7 elements presented. Because of such a poor presence of risk related concepts there couldn't be too much relationships present. Because of absence of threat agent and attack method there couldn't be any uses relationship. Vulnerability is also not present is STS, thus making impossible to model characteristic of and exploits relationships. There is no impact concept, which means that leads to, negates and provokes relationships can't be modelled. Risk concept is also absent, making significance assessed by relationship absent from STS. However, harms relationship is present in a form of *threat event* combined with *threaten* relationship. Also direction of *threaten* relationship points to asset in danger, thus modelling ISSRM targets relationship. To sum up, in scope of risk related concepts STS covers 2out of 7 elements and 2out of 9 relationships.

Risk treatment related concepts. Based on table 2 we have 1 out of 3 elements presented. Because of having only one element presented in STS we can't really build any relationships. Therefore, we have 1 out of 3 elements and 0out of 4 relationships covered in scope of risk treatment related concepts by STS.

Total. Overall, STS covers 8 out of 15 elements and 3 out of 15 relationships.

5.7.3 Integrated

Because integrated approach is a combination of STS and RSO, and they do not overlap each other (because RSO models presented on a separate view) we can say that integrated approach gains all ISSRM related elements and relationships from both STS and RSO. As for elements, RSO lacked only event concept from ISSRM, which is presented is STS in a form of threat *event*. Thus, we can say that integrated approach covers all elements from ISSRM. As for ISSRM relationships, side by side comparison of what is covered in STS and what is covered is RSO (as all these relationships are migrated into integrated approach) showed that provokes and leads to relationships remain uncovered. But, leads to relationship, which shows how event leads to impact is present in integrated approach. It is viable by connection between STS *threat event* and *risk and security view* model in a form of shared name. By this connection we can follow which event leads to which impact. Also, during integration we added *provokes* relationship from *loss event* to *loss event* (can be seen on integrated domain model), to capture ISSRM's provokes relationship. With this in mind we can say that integrated approach entirely covers ISSRM, thus making it fully complete in terms of ISSRM (15 out of 15 elements and 15 out of 15 relationships).

5.7.4 Secure Tropos

Asset related concepts. Assets are presented in a form of *resource, hardgoal, softgoal, plan* and *actor*. ISSRM *support* concept is presented in a form of *dependency, means-ends, contribution* and *decomposition* relationships. Security criterion is assembled by *softgoal* and *security constraint*, where softgoal – high security criteria and constraint – refinements it. Implicit constraint is placed by *dependency* relationship. Explicit constraint is modelled by *restricts* relationship. ISSRM Security objective is missing from Secure Tropos. Therefore, based on asset related concepts, Secure Tropos covers 4 out of 5 elements and 2 out of 2 relationships.

Risk related concepts. For this concepts we use darker backgrounds to capture malicious intentions, as it could be seen on figure 37. Therefore, when corresponding Secure Tropos constructs is mentioned in this paragraph it is meant to be with darker background. ISSRM threat agent concept is captured by *agent* concept. Attack method is modelled by *plan*. Threat is represented by *hardgoal* or *plan*. Black dot represents vulnerability. *Actor, goal, plan, targets, exploits* and *vulnerability dot* in combination represents event concept from ISSRM. ISSRM Event could also be presented as *threat* construct. ISSRM Risk is visualised by *threat* with *impacts* relationship. Impact is modelled as *impacts* relationship. As for relationships: *attack* relationship represents targets relationship. *Exploits* relationship, which links threat and asset with *vulnerability* represents ISSRM exploits relationship. Uses relationship is presented in form of agent scope (e.g. when agent executes plan which is inside his scope). Characteristic of construct is presented in a form of adding *vulnerability point* to asset as an attribute. Leads to, harms and negates relationships are presented by *impacts* relationship. However, Secure Tropos missing provokes and significance assessed by relationships from ISSRM. To sum up, based on risk related concepts Secure Tropos represents 7 out of 7 elements and 7 out of 9 relationships.

Risk treatment related concepts. To differentiate between constructs from this scope and other scopes we use dotted background as it could be seen on figure 38. Therefore, when corresponding Secure Tropos constructs is mentioned in this paragraph it is meant to be with dotted background. Security requirement ISSRM construct is modelled by combination of *actor, goal, plan, resource, softgoal* and security constraint and is visualised on security requirements definition model. Control ISSRM construct is presented in a form of combining security requirement components by *dependency, means-ends, contribution* and *decomposition* relationships. Regarding relationships, Secure Tropos models only ISSRM mitigates relationship by its *mitigates* relationship. Other ISSRM relationships (refines, implements, decision to treat) are not presented in Secure Tropos. To sum up, Secure Tropos covers 2 out of 3 elements and 1 out of 4 relationships presented in ISSRM risk treatment related concepts.

Total. Overall, Secure Tropos covers 13 out of 15 ISSRM elements and 10 out of 15 relationships.

5.7.5 DEMO

Asset related concepts. ISSRM assets, business assets and IS assets are presented as *elementary* and *aggregate production banks*. Security criterion and security objective elements are missing from DEMO. Supports relationships presented in a way of merged *access* relationships from different *production banks*. As we don't have security criterion presented in DEMO there couldn't be constraint of relationship. To sum up, DEMO covers 3 out of 5 elements and 1 out of 2 relationships in ISSRM asset related concepts.

Risk related concepts. Risk, event and threat concepts are missing from DEMO. Threat agent is modelled by *composite actor role* outside of *organizational boundary*. Attack method is described by malicious *transaction kind* connected to threat agent. Vulnerability is partially presented in textual description to *transaction kind*. Impact is partially represented by *access* links to malicious *composite actor role*, as they show what assets are impacted. Uses and exploits relationships are presented by *initiator link* from malicious *actor role* to *transaction kind*. Because of absence of event concept, it is impossible to model leads to relationship. There is no threat to target asset, thus targets relationship is absent from DEMO. Characteristic of and harms relationship is present in a form of *executor* link from malicious *transaction kind* to *elementary actor role* which connects to targeted asset by *access link*. Provokes relationship is impossible to model in DEMO, because impact concept is only partially represented. Negates relationship can't be modelled because there is no concept of security criterion in DEMO. Significance assessed by relationship can't be modelled because neither risk nor security concepts are present in DEMO. To sum up, DEMO covers 4 out of 7 elements and 4 out of 9 relationships in ISSRM risk related concepts.

Risk treatment related concepts. Risk treatment elements are completely missing from DEMO, because this language is not intended to treat risks in first place. Without any elements presented in DEMO it can't model any corresponding relationships. Therefore, we have 0 out of 3 elements and 0 out of 4 relationships coverage of ISSRM risk treatment related concepts by DEMO.

Total. Overall, DEMO covers 7 out of 15 elements and 5 out of 15 relationships from ISSRM.

5.8 Discussion

Tables below (table 7 – table 9) summarises in a more understandable manner what have been discussed in this chapter. By green colour (and ✓ symbol) we highlight that this ISSRM concept is supported in corresponding approach, by red colour (and **X** symbol) – that this concept is not supported. As it was already mentioned (and highlighted in tables) our integrated approach fully covers ISSRM concepts, whereas all other approaches miss some of the concepts. Based on this we can say that integrated approach fulfils completeness with ISSRM scope on 100%. Moreover, integrated approach was able to identify that threat of violator intercepts transfer of game report could happen in three different places, on three different transfers. Secure Tropos and DEMO wasn't able to point out this fact. Also, from this comparison it is obvious that RSO is the strongest approach (among initial ones) in risk and risk treatment related fields. This is a good sign, because we are heavily using RSO in our integrated approach. It is worth saying that Secure Tropos sometimes could give some suggestions on how attack could be done (find login to pharmacy system on technician screen, figure 37), which is missing from other approaches. In addition, it is worth mentioning that I spend more time on building integrated model than on any other model (unfortunately I haven’t measured time for building models, so I can’t give exact numbers). However, building integrated model should take at least equivalent time as building STS and RSO models, because integrated is combination of this two models. Despite the fact that integrated approach is the most time consuming, it is still worth to build it, as it gives the broadest view and the only one which fully complete in ISSRM terms.

As for huge benefit of this kind of comparison and argumentation I see lack of subjectivity. It simply couldn’t be subjective as whole comparison is based on ISSRM and we have numerical output from this comparison, which can easily point to the best solution. Regarding minuses and threats to validity of this results I would highlight possibility of mistakes in understanding some points of described approaches, which could influence results. This could cause wrong interpretation of approach concepts and as a result – change numbers of how much and what concepts from approach correlates with concepts from ISSRM. However, this possibility shouldn’t affect results for integrated approach, since it was created in this paper and is fully understandable by author. As main outcome of this comparison is fact that integrated approach fully complete in terms of ISSRM, validity threat to results of other approaches could be neglected.

Table 7. Asset related concepts comaprison

Asset related concepts						
		Tropos	STS	RSO	Integrated	DEMO
Elements	Asset	✓	✓	✓	✓	✓
	Business Asset	✓	✓	✓	✓	✓
	IS Asset	✓	✓	✓	✓	✓
	Security criterion	✓	✓	✓	✓	X
	Security objective	X	✓	✓	✓	X
Relationships	Constraint of	✓	X	✓	✓	X
	Supports	✓	✓	✓	✓	✓

Table 8. Risk related concepts comparison

Risk related concepts						
		Tropos	STS	RSO	Integrated	DEMO
Elements	Risk	✓	X	✓	✓	X
	Impact	✓	X	✓	✓	✓
	Event	✓	✓	X	✓	X
	Threat	✓	✓	✓	✓	X
	Vulnerability	✓	X	✓	✓	✓
	Threat agent	✓	X	✓	✓	✓
	Attack method	✓	X	✓	✓	✓
Relationships	Uses	✓	X	✓	✓	✓
	Exploits	✓	X	✓	✓	✓
	Leads to	✓	X	X	✓	X
	Targets	✓	✓	✓	✓	X
	Characteristic of	✓	X	✓	✓	✓
	Provokes	X	X	X	✓	X
	Harms	✓	✓	✓	✓	✓
	Negates	✓	X	✓	✓	X
	Significance assessed by	X	X	✓	✓	X

Table 9. Risk treatment related concepts comparison

Risk treatment related concepts						
		Tropos	STS	RSO	Integrated	DEMO
Elements	Risk treatment	X	X	✓	✓	X
	Security requirements	✓	✓	✓	✓	X
	Control	✓	X	✓	✓	X
Relationships	Refines	X	X	✓	✓	X
	Implements	X	X	✓	✓	X
	Decision to treat	X	X	✓	✓	X
	Mitigates	✓	X	✓	✓	X

5.9 Summary

Firstly, this chapter models the same case from perspective of five different modelling approaches. After that this chapter presents comparison criteria to compare described approaches in terms of completeness in ISSRM terms, and performs this comparison. Comparison results are summarised in tables, which highlights what ISSRM concepts are supported in each modelling approach. Taking into consideration comparison in terms of completeness from ISSRM point of view and modelled cases this chapter discusses about why our integrated approach is good providing argumentation for it, thus validating our integrated approach.

6 Conclusion

In this work we presented a brief overview of ISSRM and in details we walked through both ArchiMate RSO and STS approaches for risk management modelling. For both Risk and security overlay and STS we presented semantics, concrete and abstract syntaxes along with application guidelines and mapping to ISSRM. Based on these mappings we compared STS with RSO and identified how we can integrate them into one approach. Integration was done through combination of STS and RSO into one approach. STS approach and RSO approach remained mostly unchanged as they are not really overlapping. We combined them together as they are handling different aspects of security risk modelling. Their connection point is event concept from STS. In integrated approach event points out to threat event scope (RSO model) on newly introduced risk and security view (contains all RSO models), as RSO models extends and enhances events ability to model risks. As threat event scope and event shares the same name, we can backtrack from one to another and connect them. Integration was followed by evaluation of our new integrated approach. Evaluation is done in terms of completeness in ISSRM concepts and visual comparison of models from different approaches (STS, RSO, Integrated, Secure Tropos, DEMO). Evaluation highlighted positive sides of new approach (full completeness in ISSRM terms, pointed out on various places where considered attack is possible, heavy loaded with information risk and security view) and negative ones (the most time-consuming model building among all compared approaches).

To sum up, in this work we propose new integrated approach for security risk modelling, which is built from combination of STS and ArchiMate RSO. Integrated approach can be considered as a good one because it is fully complete in ISSRM terms, could point on specific places where attack could be done and gather risk and risk treatment related concepts on risk and security view. As a downside of integrated approach we can highlight the fact that it is more time consuming then other approaches (takes more time then STS and RSO combined).

6.1 Limitations

One of the main limitations of this work is subjectivity. Mapping of STS to ISSRM was done fairly subjectively, as it was based on our overall understanding of ISSRM and STS concepts. Also, choice of the way how STS and RSO are integrated is subjective too. This subjectivism could cause some disagreement and discussions over various points in this work, despite the fact that we tried to argument each and every decision made. Second huge limitation of this work is that example, based on which we created all our models, is not a real world example. Of course, it is based on the real world Hong Kong Blood Transfusion Centre system, but it is more like adapted for studies example. Moreover, we haven't modelled full BTC process, only a tiny part of it. That means that current integrated approach hasn't been applied to extensive real world example, and during this application there could be identified additional weak points of approach or possible improvements. Moreover, during modelling we used obvious attack methods, like SQL injection. Usage of more advanced attack method could possibly highlight some weak points in integrated approach.

6.2 Answers to Research Questions

What is ArchiMate risk and security overlay and how it maps into ISSRM?

RSO is extension to ArchiMate language, which focuses on modelling risks and security and fully relies on ArchiMate constructs with altered meaning to do this. ArchiMate in its turn is an enterprise architecture modelling language, developed by The Open Group. Detailed description of ArchiMate risk and security overlay and its mapping into ISSRM is provided in section 3.3.

What is STS and how it maps into ISSRM?

STS is a method for designing secure software systems. It builds models using its own STS modelling language. Main idea is that system design should take into consideration social interactions between actors. Detailed description for STS and its mapping into ISSRM is provided in section 3.2.

What is comparison criteria for ArchiMate RSO and STS?

As we are comparing STS and ArchiMate under ISSRM – comparison criteria will be following: to compare mappings of STS and ArchiMate into ISSRM, taking ISSRM concepts one by one and discussing their corresponding elements in STS and RSO. Moreover, we should pay attention to possible overlaps between ideas and concepts represented by elements of both approaches. In addition, during this process we should think about how concepts from both merged approaches can be merged together.

What is the optimal way to integrate ArchiMate RSO and STS?

As it was mentioned in the work, the most optimal identified way of integration STS and ArchiMate is to combine them in one approach with STS and RSO playing their corresponding separate roles. Idea was to keep original functionality of STS and RSO the same, putting RSO models into new risk and security view. This separating RSO models into new view created a question of how can we connect this view with models on other views. Solution was pretty obvious, we connected RSO models with event concept from original STS. With this connection we use RSO models as extensions and enhancements for event concept, because with a whole separate modelling approach we can model risks and security much better then with one event and threaten relationship. More details for this question is provided in chapter 4.

How can integrated approach benefit from ArchiMate RSO and STS?

From STS side – advanced threat modelling way with separate view dedicated for risk and security. From RSO side – socio-technical perspective on the system. Also, by additional adjustments (e.g. introducing provokes relationship for loss event) we achieved full coverage of ISSRM concepts in integrated approach. In details this question is answered in section 4.3.5.

How can integrated approach be evaluated?

We evaluate integrated approach in terms of completeness from ISSRM point of view. We calculate how much of ISSRM concepts is present in integrated approach. By doing this we achieve numerical result which can be easily compared with other approaches. As for other approaches (except STS, RSO and integrated) we decided to choose Secure Tropos (because it shares social perspective on modelling with STS) and DEMO (it is designed for enterprise architecture, same as ArchiMate). In addition to this we provide models for not a trivial case for all five approaches, to be able to visually compare models. All this is covered in details in chapter 5.

What is completeness of integrated approach regarding ISSRM?

Comparison provided in section 5.7 shows that integrated approach is fully complete regarding ISSRM concepts by showing how each ISSRM element and relationship is represented in integrated approach. For more details, check section 5.7.

What are the directions of future work to improve integrated approach?

Try out integrated approach on real world example, prototype implementation, revising connection between threat events and threat event scope. This question is answered in more details in section 6.3.

6.3 Future work

For future work, first of all, it is necessary to negate all identified limitations. Main work, that should be done, is to try out our integrated approach on real world enterprise example. This would prove that our subjective decisions were right or at least have a right to exist. In any case, there is nothing better for new modelling approach then feedback from real world users. Moreover, it would be good to implement prototype through which users could try out our integrated approach. Also, connection between threat events and risk and security view can possibly be improved.

7 References

[1] Dalpiaz, F., Paja, E., & Giorgini, P. (2016). Security Requirements Engineering: Designing Secure Socio-Technical Systems (p. 224).

[2] NIST. (2013). Glossary of key information security terms. NIST IR (Vol. 7298). Retrieved from http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

[3] Russell, N., A. H. Ter Hofstede, D. Edmond, and W. M. van der Aalst (2004). Workflow resource patterns. Technical report, BETA Working Paper Series, WP 127, Eindhoven University of Technology.

[4] Band, I., Engelsman, W., Feltus, C., Paredes, S. G., Hietala, J., Jonkers, H., & Massart, S. (2014). Modeling Enterprise Risk Management and Security with the ArchiMate Language. Open Group, 40.

[5] Iacob, M.-E., Jonkers, H., Lankhorst, M. M., Proper, H. A., & Quartel, D. A. C. (2012). ArchiMate 2.1 Specification; available at: www.opengroup.org/bookstore/catalog/c13l.htm.

[6] Mayer, N., & Feltus, C. (2017). Evaluation of the risk and security overlay of archimate to model information system security risks. In Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW(Vol. 2017–October, pp. 106–116). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/EDOCW.2017.30

[7] The Open Group (2017). ArchiMate 3.0.1 Specification. ArchiMate 3.0.1 Specification.; available at: http://pubs.opengroup.org/architecture/archimate3-doc/toc.html

[8] University of Trento. (2014). STS-ml Modeling Language Manual; available a: http://www.sts-tool.eu/manuals/

[9] Matulevičius, R. (2017). Fundamentals of secure system modelling. Fundamentals of Secure System Modelling (pp. 1–218). Springer International Publishing. https://doi.org/10.1007/978-3-319-61717-6

[10] University of Trento. (2014). STS modeling language tutorials; available at: http://www.sts-tool.eu/tutorials/

[11] The Open Group. (2013). Risk Taxonomy (O-RT), Version 2.0, Open Group Standard (C13K); available at: www.opengroup.org/bookstore/catalog/c13k.htm.

[12] Tovstukha, I. (2014). Management of Security Risks in the Enterprise Architecture using ArchiMate and, 1–53.

[13] Mouratidis, H. (2004). A Security Oriented Approach in the Development of Multiagent Systems: Applied to the Management of the Health and Social Care Needs of Older People in England

[14] Mouratidis, H., & Giorgini, P. (2007). Secure Tropos: A Security-oriented extension of Tropos methodology. International Journal of Software Engineering and Knowledge Engineering, 17(02), 285–309; available at: https://doi.org/10.1142/S0218194007003240

[15] Mouratidis, H., Giorgini, P., & Manson, G. (2003). Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. Proceedings of CAiSE 2003, 15th International Conference on Advanced Information Systems Engineering, 2681, 63–78.

[16] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems, 8(3), 203–236; available at: https://doi.org/10.1023/B:AGNT.0000018806.20944.ef

[17] Kolk, K. (2015). An Empirical Comparison of Approaches for Security Requirements Elicitation.

[18] NIST. (2017). Framework for Improving Critical Infrastructure Cybersecurity V. 1.1. National Institute of Standards and Technology, 1–41.

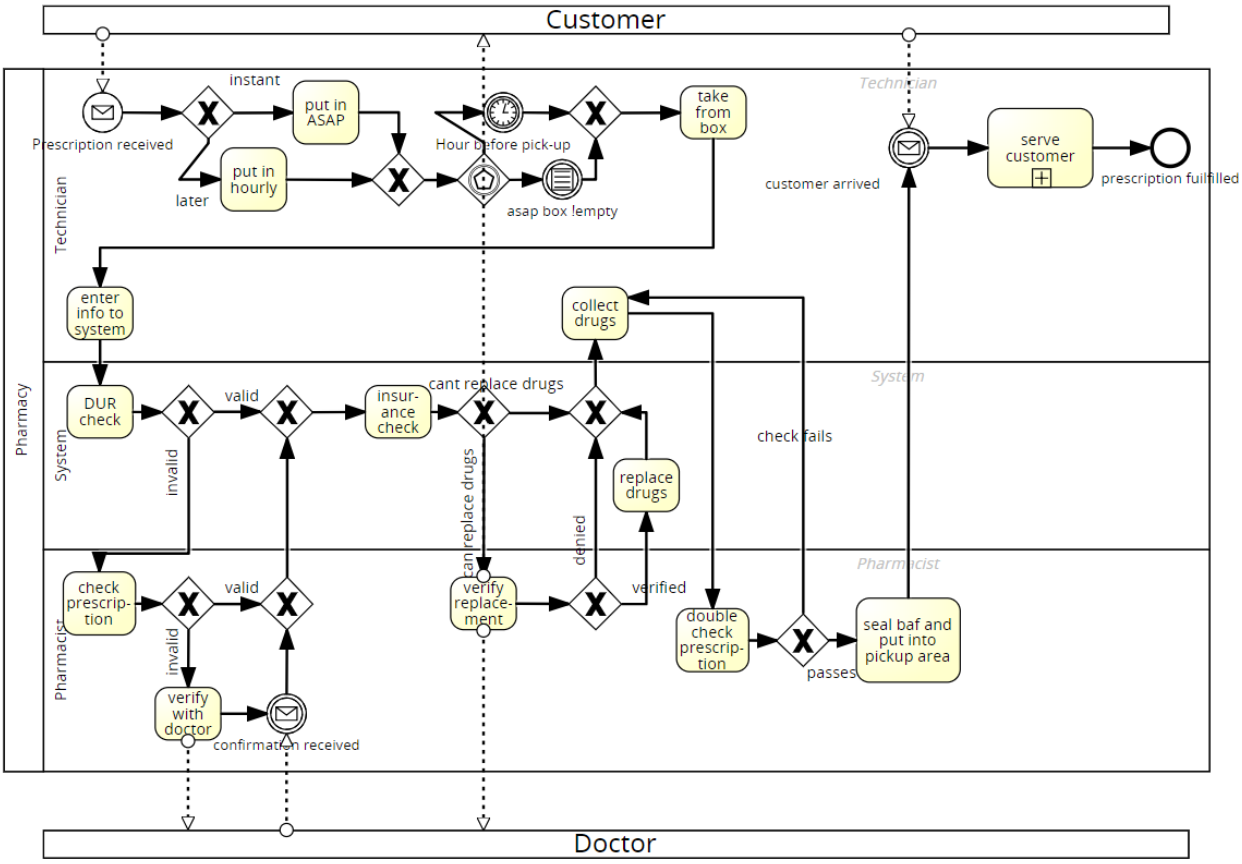
[19] Landess, D. (2003). A Practical Information Risk Management Process Framework; available at: https://www.giac.org/paper/gsec/3303/practical-information-risk-management-process-framework/105444

[20] Dietz, J. (1999). Understanding and Modeling Business Processes with DEMO. Conceptual Modelling - ER '99, LNCS 1728, 188–202. https://doi.org/10.1007/3-540-47866-3_13

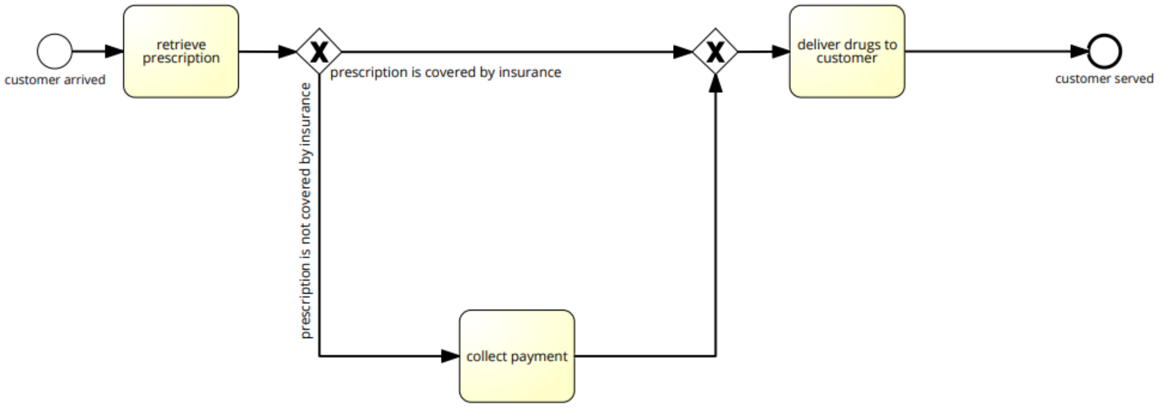
[21] Enterprise Engineering Institute. (2017). DEMO specification language 3.6; available at: http://www.ee-institute.org/en/documents/21/methodology-documents

Appendix

I. Prescription fulfillment process



II. Serve customer from prescription fulfillment process


customer arrived
retrieve prescription
prescription is covered by insurance
prescription is not covered by insurance
collect payment
deliver drugs to customer
customer served

III. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Zaitsev Artem**,

(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Comparison of STS and ArchiMate Risk and Security Overlay,

(title of thesis)

supervised by Raimundas Matulevičius,

(supervisor's name)

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **24.05.2018**