# Automated Architecture Modeling for Enterprise Technology Management Using Principles from Data Fusion: A Security Analysis Case

Margus Välja[1], Matus Korman[1], Robert Lagerström[1], Ulrik Franke[2], and Mathias Ekstedt[1]
[1]KTH Royal Institute of Technology, Stockholm, Sweden
[2]SICS, Swedish Institute of Computer Science, Stockholm, Sweden

*Abstract*--**Architecture models are used in enterprise management for decision support. These decisions range from designing processes to planning for the appropriate supporting technology. It is unreasonable for an existing enterprise to completely reinvent itself. Incremental changes are in most cases a more resource efficient tactic. Thus, for planning organizational changes, models of the current practices and systems need to be created. For mid-sized to large organizations this can be an enormous task when executed manually. Fortunately, there's a lot of data available from different sources within an enterprise that can be used for populating such models. The data are however almost always heterogeneous and usually only representing fragmented views of certain aspects. In order to merge such data and obtaining a unified view of the enterprise a suitable methodology is needed. In this paper we address this problem of creating enterprise architecture models from heterogeneous data. The paper proposes a novel approach that combines methods from the fields of data fusion and data warehousing. The approach is tested using a modeling language focusing on cyber security analysis in a study of a lab setup mirroring a small power utility's IT environment.**

## I. INTRODUCTION

The interplay between technology and business operations in modern organizations is becoming increasingly complex. Organizations need to utilize their assets in best possible ways in order to fulfill their missions. The discipline of Enterprise Architecture (EA) offers methods and models to aid organization in achieving this goal.

To create EA models, different kinds of information about processes and architectural elements are needed. For a bigger organization with many business units, processes, and vast architectural details, creating EA models manually often becomes an overwhelming task. While there are some studies that explicitly measure the time and effort needed to carry an EA analysis out (e.g. [24]), this is the exception rather than the rule. Accuracy of the models can also become a problem, as architecture tends to change over time. Farwick et al. [7] have seen manual data collection and ensuring sufficient quality as major challenges in enterprise architecture.

When planning for architectural changes timely information is needed about the current status. There are a variety of architectural aspects that this is true for, like business process management, IT governance [28], interoperability [37], availability [13] and modifiability [21]. Timely information is probably most essential when it comes to security analysis and making informed security investment decisions. Cyber security is a dynamic and quickly changing field. Serious security problems could cause an enterprise to close down business services, and failure to meet legal obligations can result in huge fines and damage to reputation.

The practice of operational security recognizes this, using intrusion detection systems (IDS) and security information and event management (SIEM) systems to keep track of what is going on in computers and networks. Indeed, such cyber situational awareness with a focus on incidents and security is a field that has received a lot of attention lately [12]. However, strategic security decisions in enterprises are still far too often based on outdated PowerPoint slides and models that can only be updated manually, despite the fact that security models need to be updated frequently and the quality of data for model creation must be maintained. Due to the overwhelming work of keeping track of strategically relevant security attributes of organizational assets, automation of modeling becomes especially important. Making use of automatic data collection tools and methods can help to overcome the labor-intensive data collection phase and provide a more accurate and updated overview of the strategic situation.

There have been attempts to automate modeling before. The authors succeeded in creating two different types of models from one dataset [4; 14]. Alegria and Vasconcelos created a logical inference framework that using raw network traffic was able to reason over information system architecture models [1]. However, a single source of data is rarely enough for describing IT architecture. Models used for analysis and decision making often need data from more than one source as they can span over multiple domains. This was also corroborated by our earlier work [39].

Consolidating operational data from multiple data sources is not a new topic. It has been studied as part of military and civilian systems in the field of data fusion [22] for some time. Data fusion deals with association and combination of data to assess situations and their importance in a timely manner. This knowledge is what we believe is needed when modeling with multiple data sources for enterprise architecture, enterprise IT architecture, and strategic cyber security management. Data fusion already plays an important role in maintaining operational, though not strategic, security situational awareness.

In this paper we build on a well-known data fusion framework named after Joint Directors of Laboratories (JDL) [2] to solve the problem where a complex enterprise IT architecture model needs data from several sources. In our case the focus is on cyber security modeling and analysis, although the approach is applicable to other related models as well, such as [17]. To do automatic modeling, the data needs to be collected, processed, and used without (or with very little) human involvement.

The paper is divided into seven sections. Section 2 gives an overview of the related work and introduces important concepts that are needed to understand the contribution. Section 3 describes the proposed approach and section 4 contains an empirical study using this approach. The discussion can be found in section 5, and section 6 concludes the paper.

## II. RELATED WORK

Automatic creation of enterprise (IT) architecture models has received some, but not much attention in research. The literature seems to be lacking in practical approaches that use more than one heterogeneous data source for modeling. However, heterogeneous data sources are an important topic in data fusion where they are used for prediction and situation awareness. The following subsections introduce enterprise modeling and data fusion.

### *A. Enterprise modeling*

Farwick et al. [10] did a systematic literature review and found 28 sources dealing with EA data collection. Eight categories of data collection where identified including, 1) interviews & forms, 2) wiki collaboration, 3) defined data collection processes, 4) generic import concepts, 5) tool-, model-, semantic integration, 6) automation via specific data sources, 7) change events & notifications, and 8) conflict resolution & quality assurance. As expected, most deal with manual collection.

Moser et al. [23] propose enterprise architecture management patterns as a supplement to existing frameworks. One of the patterns proposed relates to automatic data acquisition and maintenance. The participants of the pattern are identified as a data provider, domain experts, and enterprise architects. The authors name transformation rules as a major challenge. The authors also state that an enterprise architecture management tool needs to be able to import data from multiple sources.

Semi-automated processes for maintaining enterprise architecture models are introduced by Farwick et al. [7]. The processes are designed to collect data from technical interfaces with the help of humans. The processes are based on an earlier work of requirements for a tool that would support semi-automated EA maintenance. The technical interfaces in the paper are assumed to be web services and the internal data structure is assumed to be machine-readable. They also state that EA data should be collected from various information sources.

Farwick et al. [9] propose a meta-model for automated enterprise architecture model maintenance that according to authors supports recurring data collection and maintains imported elements to their sources. The proposed meta-model stays on a high level and the authors do not explain the details of their identity reconciliation and information fusion techniques, but only state that the data of identified duplicates is merged.

Data quality and merging issues can be addressed with data cleaning methods. Rahm and Do [27] classify data cleaning problems and give an overview of data cleaning approaches. Their topic of interest is data quality and they investigate how to improve data quality when integrating data from heterogeneous sources. The authors state that since information from data warehouses are used for decision-making, data correctness is vital. This is also the case with EA models.

The authors identified two previous attempts to automate EA modeling. The first attempt is by Buschle et al. [4] where the authors automatically created a model, using a security metamodel called CySeMoL, with data from a vulnerability scanner Nexpose. Holm et al. use the same data source to generate an ArchiMate based EA model [14]. Both models were created using a single source of data and only partially populated. Data integration was not in the scope of either paper. It is clear that to populate complex models, data from multiple data sources needs to be combined and thus data integration problems must be solved.

### *B. Data fusion*

Steinberg [31] defines data fusion as a process where data or information is being combined to predict or estimate states of an entity. The field addresses problems such as data alignment, association, and estimation. Fusion is widely used as part of military and civilian systems. Hall and Llinas [22] name some applications as strategic warning and defense, ocean surveillance, and robotics. Fusion is used in order to make timely decisions and often offers improved estimates and statistical advantages over data from single sensors.

Elmenreich [6] describes two types of fusion, information and sensor fusion. Sensory fusion encompasses only combining and processing sensory data, while information fusion covers a broader range of data sources. These sources might be databases, information from experts, and data from sensors. Elmenreich further divides fusion into three levels; low-level, intermediate, and high-level fusion. Low-level fusion is used to combine raw data sources, intermediate level fusion combines features like lines and textures, and high-level fusion combines decisions with voting, fuzzy logic, or statistics. Elmenreich sees sensor configurations as complementary, competitive, or cooperative. A complementary setup helps achieving greater completeness of data, a competitive one can improve the reliability and accuracy, and a cooperative one can improve the derivation of information.

The understanding of natural language in data fusion is seen as a problem of situational assessment. Steinberg [33] defines situation assessment as estimation and prediction of parts of reality that involves inferring the presence, state, relationships of entities, recognizing and characterizing situations, and predicting unobserved situations. The uncertainty of the beliefs of an information system needs to be represented by an uncertainty metric. According to Steinberg, context is used to improve ambiguous estimates, explain observations, and limit processing. The dependencies between problem and context variables are then represented using factor graphs where situation is given as a sub-graph. Real world information about e.g. physical or geo-political surroundings might be usable for this purpose.

A well-known fusion framework Joint Directors of Laboratories (JDL) describes data fusion on five levels [22; 32]. In the JDL framework association, correlation, and combination of data and information are used for purposeful and iterative refinement process. The levels in the JDL model are designed to help with categorization of logically different types of problems. Kessler and White [22] state that the JDL model could be used for partitioning functions and as a checklist for fusion capabilities in information or decision support applications. Other authors have use the framework for information exploitation [2].

In this paper we build on the JDL framework and study data fusion to create a process for automating model creation from multiple heterogeneous sources.

## III. AUTOMATING WITH FUSION

In this section we present a modeling approach for enterprise IT modeling that builds on data fusion research. The approach combines information from multiple enterprise IT data sources. In the first subsection we introduce possible data sources and in the second our proposed approach.

### *A. Enterprise IT data sources*

Moser et al. [23] and Farwick et al. [7] point out the need to acquire data for enterprise architecture management from multiple sources. In addition to the systematic literature review [10] mentioned in section 2, there are other studies looking at enterprise architecture data sources. Farwick et al. [8] conducted a survey among enterprise architecture practitioners and gathered the main sources for enterprise architecture management information. They mention network monitors and scanners, configuration management databases, project portfolio management tools, enterprise service bus, change management tools, license management tools, directory services, business process engines, and release management tools. Many of the aforementioned tools are also found to be good sources of enterprise architecture change events [11]. A non-comprehensive list of potential enterprise data sources is provided in Table 1. It is important to consider that some data collection methods like fingerprinting network responses are less trustworthy due to the uncertainty involved as compared to exporting configuration data directly from a configuration management system.

### *B. Process*

The purpose of the proposed automation process is to be able to create accurate enterprise IT architecture models using heterogeneous enterprise data sources in a cost effective way. The process we introduce here can only be used to create models that have a predefined ontology. The automation of the modeling is achieved partially.

The automation process builds on five JDL levels that are explained in Table 2 and the actual automation needs manual preparation. This manual preparation should take place in several iterations to achieve the desired quality level. The quality assurance here is up to the persons who are putting in the effort to establish the automation. Fig.1. shows how the automation is achieved and related JDL levels. There are three steps that need to be manually iterated. These steps are shown in Fig. 2 and explained as follows.

TABLE 1. ENTERPRISE DATA SOURCES FOR MODELING.

| **Type of tool** | **Examples** | **Type of data** | **Data acquiring method** |
|---|---|---|---|
| Active scanners | Vulnerability scanners, network scanners | Hardware devices, software, vulnerabilities | Scanning network, computer nodes and application servers |
| Passive scanners | Vulnerability scanners, network scanners, packet analyzers | Hardware devices, software, vulnerabilities, network communication | Listening to existing network traffic |
| Enterprise architecture management | Business, information, IT architecture | Models of organization and its IT (in different views) | Manual input, scanning |
| System management | Change, release, license management, directory services | TO-BE to AS-IS elements | Manual input, scanning |
| Security monitors | IDS, IPS, firewalls, SIEM solutions | System, network, process state information | Scanning, listening, registering security events |

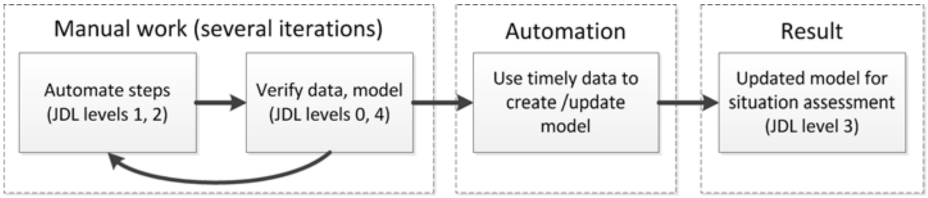
Manual work (several iterations)
Automate steps (JDL levels 1, 2)
Verify data, model (JDL levels 0, 4)
Automation
Use timely data to create /update model
Result
Updated model for situation assessment (JDL level 3)

**Fig. 1.** Manual work to achieve automation.

TABLE 2. DATA FUSION MATCHED TO MODELING AUTOMATION [22; 31].

| JDL Fusion level | JDL tasks | Modeling automation |
|---|---|---|
| Level 0: Source preprocessing/sub-object refinement | Precondition to correct biases, align data, standardize inputs | Standardizing data sources |
| Level 1: Object refinement | Association of data to estimate an object or entities position, or attributes | Identifying objects, attributes, associations |
| Level 2: Situation refinement | Aggregation of objects/events to perform relational analysis and estimation in context | Comparing standardized data across multiple sources |
| Level 3: Impact assessment | Projection of the current situation to perform event prediction, consequence analysis | Role filled by the model |
| Level 4: Process refinement | Evaluation of the ongoing process to provide advisories, fusion control and request additional data | Improving the automation process, alternatively the model itself. See Fig. 1. |

The first step is to create two data structures based on the model ontology (metamodel). Two data structures are needed because they fulfill two different purposes. The aim of the first data structure is to facilitate data to model transformation, thus it must closely resemble the model ontology. The aim of the second data structure is to facilitate integration of data from multiple data sources and also to facilitate quality assurance. The second data structure is therefore an extended version of the first one that includes metadata for analysis and integration. A major difference between the data structures is the level of granularity. The first data structure represents the objects of the model ontology while the second data structure the key parts that make up the ontology elements and that can be found in the available data. For example a network zone in data structure 1 could be represented by a subnet address and a department name in data structure 2.

The second manual step as shown in Fig. 2 is to pick enterprise data sources for model creation and to create an adapter for each of them. These adapters should translate the data from the source's data structure to the second data structure that was created in the first step. Metadata, such as the name of the source, time of acquiring the data and user trustworthiness in the data source need to be included, to maintain quality and traceability during the integration process. It should be noted that only a partial representation of the data structure 2 (extended data structure) can be created for each data source, because every data source contains different amount and types of data. To manage this, each adapter should also create metadata annotations about what type of data is available in each instantiation of data structure 2.

A crucial task for the adapters is to standardize the data before completing step 2. Standardization involves converting data to a uniform format, for example dates and addresses. This includes using external data from inside and outside the organization to decide for example how department names and addresses should be used. The other data processing steps that can be taken with the help of adapters are removing duplicates, extracting values, and validation. Value extraction here means reordering the values so that they can be compared with each other. Validation means identifying data errors using dictionaries or known dependencies, such as comparing total price to unit price times quantity.

Once the adapters have been created, the actual data integration needs to be set up (step 3). During the data integration, that we also call analysis, multiple representations of data structure 2 are transformed into a single representation of data structure 1. This is the data that now can be used for data to model transformation using established methods like XSLT [38]. Some of the possible data processing tasks on both adapter and analysis level are described in Table 3.

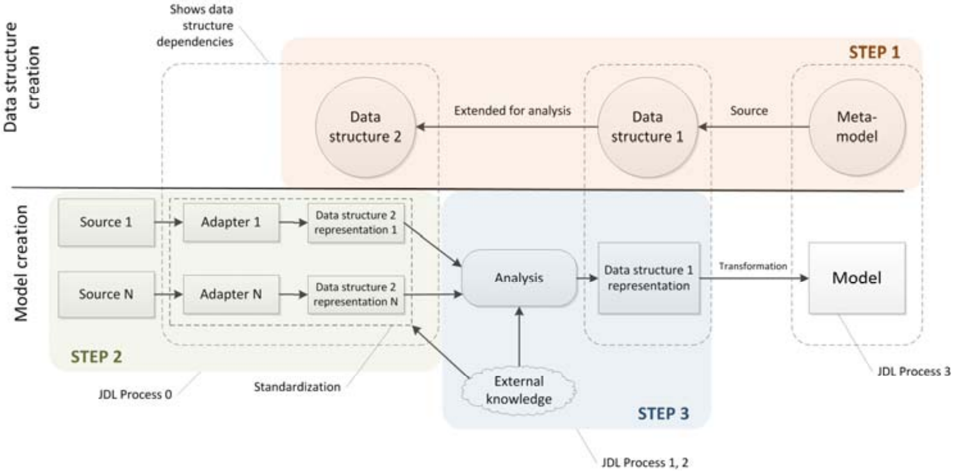

**Fig. 2.** Three steps to accomplish automation.

TABLE 3. DATA PROCESSING TASKS. PARTS TAKEN FROM [27].

| Level | Goal | Method | Example |
|---|---|---|---|
| Adapter | Finding relevant data, including data extraction | Match to data structure 2 | Assets, associations |
| Adapter | Remove duplicates | Find unique matches | Removing info about repeating packets |
| Analysis | Find associations between data elements (resolving conflicts) | Link data elements using unique identifiers | Link software using IP address |
| Analysis | Find associations between data elements (resolving conflicts) | Link using fuzzy matching (letter, word) | Link software across data sets using parts of name |
| Analysis | Add world data | Use dictionary, ontology | Add port names to numbers |
| Analysis | Reducing the size of model | Use templates (reference models) | Exclude the list of Linux system libraries from a list of software identified |
| Analysis | Recognize patterns | Recognize combinations of certain data elements as an object or an association | IPtables installed in Linux as a software firewall |
| Analysis | Check model for validity | Compare against known patterns | Check if two or more data elements that are supposed to have an association, have one |

Setting up data analysis needs some effort. Important goals for the analysis are finding associations and resolving conflicts. Data from outside sources (external knowledge) might be required to fill in gaps that appear when comparing different data representations with one another. For example communication ports might be represented by numbers in one data set and names in another.

The first task of the data analysis is to find if each representation of data structure 2 that was created with the help of an adapter contains sufficient amount of data to create data structure 1 ontology objects. As stated previously, each data structure 1 ontology object needs certain key data (combination of values) to be created. The information about the key data should be available as annotations in the metadata for each imported data source and were created during the data structure 2 representation creation process. Once it is clear what ontology objects can be created for each data source, the next step is linking them together by finding associations between them like for example IP addresses.

Data from multiple sources can complement each other and raise the accuracy of the models that are created using them. However, some of the data might be overlapping and contain conflicting information. Solving those conflicts is the next major task. Data fusion recognizes three types of fusion, competitive, complementary and cooperative [6]. The goal of the competitive fusion is to improve reliability and accuracy of the same type of data by choosing the most trustworthy source. The role of the complementary fusion is to improve completeness of the data if the data sources contain heterogeneous data and the cooperative fusion in the same case aims to infer a completely new type of data. The three types of data fusion steps can be accomplished with the help of simple statistical methods and enterprise data source trustworthiness score (set by the user for each source). Statistical methods can be for example used to calculate the most frequent result if many data sets contain data about the same real world entity, or alternatively only data from the most trustworthy data set can be taken and the rest ignored. There are other methods.

The goal of the modeling automation is to create a process that is repeatable with minimal manual work. In case of large amounts of data, modeling automation might be the only way to create a model at all. A programming language can be used as a facilitator of automation, so that whenever updated data becomes available, it is just a matter of running a program, or a series of scripts, to update the model.

## IV. PRACTICAL STUDY

In the following section we describe an effort to automate the creation of an enterprise IT architecture model using three data sources in a lab environment. The purpose of the attack graph based model is to predict cyber security threats.

### *A. EAAT and CySeMoL*

We have developed a tool for enterprise IT architecture modeling and analysis. The tool is called the Enterprise Architecture Analysis Tool [4; 18]. It is based on the Unified Modeling Language (UML) [25], and an extended version of the Object Constraint Language (OCL) [26] called P$^2$AMF [16]. The tool can be used for modeling of two types of models - metamodels (also called class models) [20] and instance models (also called object models). A metamodel contains a representation of a certain modeling domain, represented as UML class diagrams, and the corresponding evaluation logic, implemented in OCL.

One example of such a metamodel is P$^2$CySeMoL [15], another example is MAP [17]. Our automatic modeling effort in this paper focuses on populating a P$^2$CySeMoL model. P$^2$CySeMoL is an extension of the Cyber Security Modeling Language (CySeMoL) [29] and predicts the probability with which a single attacker or multiple attackers can compromise different parts of the architecture, based on its structure and the properties of its parts (e.g., the attributes of the systems the architecture consists of). It is an attack graph based cyber security evaluation model that assumes that the attacker is a professional penetration tester having access to any publicly available tools that support performing cyber attacks. In this paper, P$^2$CySeMoL is also referred to as just CySeMoL.

The overall data needs for CySeMoL are as follows. The needs are here divided into three categories; Collected, Known, and Unknown. With the Collected category we show what data has already been collected during previous studies [3; 4; 14]: Application protocols, Computer and network hardware with addresses, Network zones, Software (also firmware) including system software and operating systems, User accounts.

The Known category lists the data needs that the authors of this paper have studied and have a solution for: Known vulnerabilities in existing software, Patch levels of clients, servers and software products, Access control points and password authentication mechanisms, Data flows.

The Unknown category contains a list of needs that are the focus of future studies: Configuration methods used for web applications and similar, IT management processes' characteristics like for example for zone management process, Social aspects like social zone, security awareness program, and developer training, Software architecture and software assurance methods like static code analysis, Types of security controls present like cryptography methods and port security.

### *B. Lab setup*

The focus of this study is a supervisory control and data acquisition (SCADA) lab with five special purpose servers that are running various operating systems e.g. Windows Server 2003 and Red Hat Enterprise Linux. The lab was created as part of EU financed VIKING project [19] to test SCADA equipment. The servers are not regularly maintained, meaning that the number of known vulnerabilities steadily grows over time.

Three different tools are chosen to gather data about the lab environment. The first choice is Nexpose, an active network scanner, which in previous studies has been found to be accurate in comparison with similar software [30]. The second choice is to record network traffic passively using a well-known network traffic analyzer software called Wireshark [5]. Passive scanners are especially useful in environments where probing a network environment actively by generating network traffic might cause disturbances. The third pick is Nessus [35], another network vulnerability scanner that was chosen as a comparison to other tools.

For data processing Talend Open Studio [34] was selected because it allows quick prototyping, there is a free version, and it has extensive documentation available. A Postgres SQL database [36] was used as a data repository and as a secondary data processing tool.

### *C. Implementation*

The modeling effort focuses on populating the CySeMoL attack graph based cyber security evaluation model using three data sources. The goal of the effort is to automatically generate model elements and demonstrate the process proposed in the Section 3.2 in practice.

The first step in the proposed approach is the creation of two data structures. The first data structure is the representation of the ontology of the modeling language, CySeMoL. The purpose of this data structure is to allow easy data-to-model transformation. In our case the data structure represents the elements in the ontology (metamodel), which are classes, groups of classes (templates), attributes, and associations between the classes and the templates in CySeMoL.

The second data structure is an extended version of the first data structure. It includes metadata for supporting analysis and maintaining data quality. The goal of the second data structure is to first support data standardization with the help of adapters and secondly to support data consolidation. The aspects that are considered here are automatically generated annotations that characterize the content (ontology key elements) of each data set, trustworthiness values for each data source set by the user, and the time the data was obtained. In our case we have three data sources which are Nexpose, Nessus and Wireshark. We also might want to calculate the trustworthiness of different pieces of data using statistical algorithms. However, the exact calculation algorithms are not in the scope of this article. An example for trustworthiness would be if a user might consider data from Nexpose to be more trustworthy than Nessus, then Nessus data would only be used to complement Nexpose data. However, if both data sources are seen equally trustworthy, other steps need to be taken to consolidate the data from both.

Once the data sources have been chosen and data structures have been designed, it is time to focus on data transformations. In the first round of transformations, adapters for each data source were implemented with Talend, the result being three representations of data structure 2 with the appropriate annotations generated to characterize the sets. One representation is for Nexpose, one for Nessus and one for Wireshark. These representations are then used in the second transformation round (analysis) to create a single representation of data structure 1. The second transformation round consists of a series of competitive, complementary and cooperative fusion steps. Most important analysis steps which are applicable for our implementation are described in table 3 together with CySeMoL specific examples.

During the second transformation round we need to merge three sets of data that represent different aspects of the SCADA lab. While the network scanners Nexpose and Nessus give us a list of assets and software, the network traffic capturing software Wireshark gives us interactions between these assets. Our goal is to merge the different data sets so that they complement each other. For that purpose we need to find common unique identifiers that can be used to link the sets to each other. In our case the main unique identifier is the network address. The network address can link each asset with certain software with a particular data flow. Physical address may be used to complement the network address if similar (internal) network addresses are used in different locations. Other less unique identifiers included software name and version, and endpoint name.

There were 5 adapter level problems we had to address. Two examples are the following. There was a difficulty with comparing operating system names from Nexpose and Nessus. The Nexpose XML export file had separate tags for software names, vendors and versions, while in Nessus' case this data had been merged together and had to be extracted. This had to be done on adapter level. Another adapter level problem was related to data abstraction. Wireshark gives us every small interaction between network nodes, but only application level protocol based data interactions are relevant for CySeMoL. To abstract Wireshark data, we filter out everything except unique data elements that contain application level protocol info.

There were 7 analysis level problems and here we present 3 as follows. An example of a context related issue is that we get a list of software from our sources, but we don't know which software instances are acting as servers. For that purpose a comparison of the list of identified software on each endpoint against the known server programs from external sources is needed. We found that this distinction needs to be done already on the adapter level before any data structure 2 software elements have been created. Another issue was to match the known server programs to identified open ports and traffic on our endpoints, because the service and server software names differed. The third problems was that Nexpose picked up around 700 different instances of software on two Red Hat servers, while only 20 were found for Windows servers. This difference comes from what is counted as operating system internals and what is not. For example Linux systems packages like xorg, gnome and kernel were also listed as external software. We needed to store the names of these system packages and exclude them from further analysis.

The final part of the approach for automatic modeling is to go through the actual data-to-model transformation. The input to the transformation is a data file that we generate as the result of our data processing. This data file has the first data structure that already resembles the structure of the CySeMoL metamodel. As the last step we need to create the transformation file that maps the CySeMoL elements to our data one-to-one. In this study, the file we generate as a data source is of XML format. The transformation file is written in XSLT, as this is the only import functionality supported by EAAT (the modeling tool used in this study). Once we have both files ready, we can invoke the transformation process in EAAT.

*D. Results*

By following the steps inspired by the JDL data fusion model and applying data cleaning and integration techniques we were able to create an automation process that is able to merge multiple data sets into one data set and thus allow us to create and update the CySeMoL model of our lab with little effort. Although manual work was used to set up the actual process, modeling the lab manually would have been equally time and effort consuming and not that accurate. Altogether over 20 000 objects were generated, resembling the details like installed software packages and data flows. The objects are grouped into templates to be comprehensible and a small part of the model is shown in Fig. 3.

To predict the (security) events and consequences as described in JDL framework level 3 we need to initiate the actual enterprise IT security analysis using the model we created. This step is outside the model automation process as such.

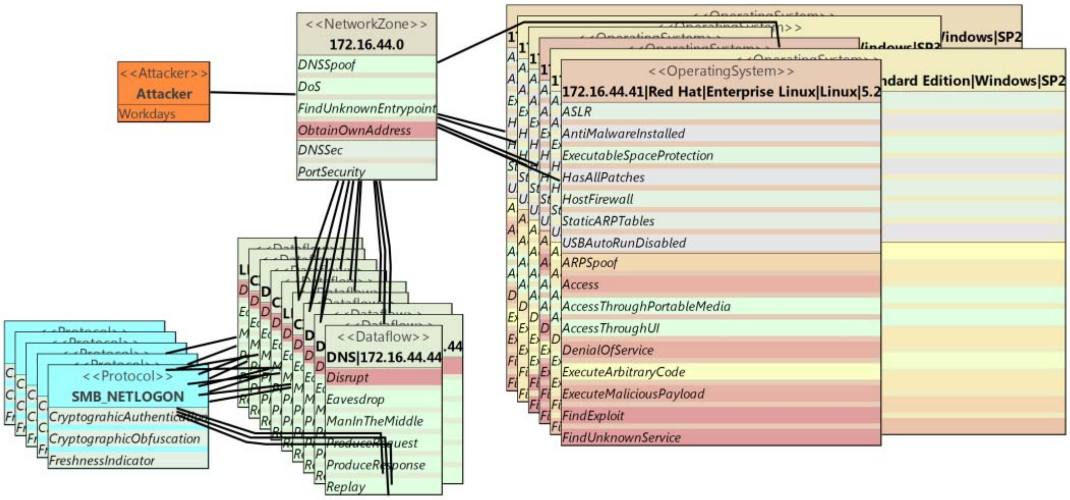

**Fig. 3.** A small part of the CySeMoL model with instantiated templates. Out of 20 000 objects in the model, 60 are shown in this screenshot.

## V. DISCUSSION AND FUTURE WORK

Enterprise IT architecture models are used by various enterprise stakeholders to communicate with each other. Even if the models cover only limited aspects of an enterprise, they can become large and complicated. Therefore it makes sense to automate model creation as much as possible. However, if multiple data sources are involved then data quality issues become especially pressing and need to be addressed in the best possible way. By building on the established field of data fusion where data quality problems have been investigated, we are able overcome this problem.

The work presented in this paper has some limitations. The first one is that although the main topic of this paper is automatic model creation, the actual generation of the model from the merged data is not explained. The reason for this is that model to model generation has been studied by other authors in earlier research and there are known methods for this like XSLT.

Another limitation is that the paper explains the general process of automation, but does not explain the data analysis methods in detail. The main reason for that is that the analysis steps are ontology (metamodel) specific. In our case we created a separate workflow for each element type in our model and applied techniques such as statistical frequency calculation, trustworthiness comparison and fuzzy text matching. The goal of the future work is to standardize analysis steps as much as possible and to create formalisms that allow calculating probabilistic values as data quality measures to the individual objects after successful data integration.

An interesting question for the future is what additional infrastructure knowledge might make the whole process simpler (e.g., repositories of common patterns, reference models, etc.). This also will be part of future work.

## VI. CONCLUSION

In this paper we have presented an approach to automate enterprise IT architecture modeling using multiple data sources. The approach builds on a well-known JDL data fusion framework and data warehousing techniques.

The core of the approach relies on five JDL framework levels to generate an enterprise IT model for strategic decision support. The enterprise IT model itself fulfills the role of the situational analysis and prediction level. The goal is achieved by standardizing data structures, keeping track of data quality related metadata and using external sources to complement gaps between the data sets. The technique involves object identification and situation assessment where each object's properties and associations are evaluated. The data are combined using competitive, complementary and cooperative techniques.

To demonstrate the feasibility of the approach, the paper includes a study. In the study three data sources are used to generate an enterprise IT cyber security model. It is shown with the study that it is possible to automatically generate timely and scalable enterprise IT models from heterogeneous data.

## ACKNOWLEDGMENTS

This project has received funding from the European Union's Seventh Framework Program for research, technological development and demonstration under grant agreement no. 607109 (SEGRID). Also, part of it has been financed by Svenska Kraftnät (http://www.svk.se).

## REFERENCES

[1] Alegria, A; Vasconcelos, A., "IT Architecture automatic verification: A network evidence-based approach," 2010 4th Int. Conf. Res. Challenges Inf. Sci. - Proceedings, RCIS 2010, pp. 1–12, 2010.

[2] Blasch, E.; Steinberg, A; Das, S.; Llinas, J.; Chong, C.; Kessler, O.; Waltz, E.; and White, F., "Revisiting the JDL model for information exploitation," Proc. 16th Int. Conf. Inf. Fusion, FUSION 2013, pp. 129–136, 2013.

[3] Buschle, M.; Ekstedt, M.; Grunow, S.; Hauder, M.; Matthes, F.; and Roth, S., "Automating enterprise architecture documentation using an enterprise service bus," Am. Conf. Inf. Syst., 2012.

[4] Buschle, M.; Holm, H.; Sommestad, T.; and Ekstedt, M., "A tool for automatic enterprise architecture modeling," CAISE11 Forum, p. 8, 2011.

[5] G. Combs, "About Wireshark." [Online]. Available: https://www.wireshark.org/about.html. [Accessed: 02-Feb-2015].

[6] Elmenreich, W., "Sensor Fusion in Time-Triggered Systems," Technische Universität Wien, 2002.

[7] Farwick, M.; Agreiter, B.; Breu, R.; Ryll, S.; Voges, K.; and Hanschke, I., "Requirements for automated enterprise architecture model maintenance a requirements analysis based on a literature review and an exploratory survey," 13th Int. Conf. Enterp. Inf. Syst. (ICEIS), Beijing, pp. 325–337, 2011.

[8] Farwick, M.; Breu, R.; Hauder, M.; Roth, S.; and Matthes, F., "Enterprise architecture documentation: Empirical analysis of information sources for automation," 2013 46th Hawaii Int. Conf. Syst. Sci., pp. 3868–3877, 2013.

[9] Farwick, M.; Pasquazzo, W.; Breu, R.; Schweda, C.M.; Voges, K.; and Hanschke, I., "A meta-model for automated enterprise architecture model maintenance," Proc. 2012 IEEE 16th Int. Enterp. Distrib. Object Comput. Conf. EDOC 2012, pp. 1–10, 2012.

[10] Farwick, M.; Schweda, C.M.; Breu, R.; and Hanschke, I., "A situational method for semi-automated Enterprise Architecture Documentation," Softw. Syst. Model., 2014.

[11] Farwick, M.; Schweda, C.M.; Breu, R.; Voges, K.; and Hanschke, I., "On enterprise architecture change events," Lect. Notes Bus. Inf. Process., vol. 131 LNBIP, pp. 129–145, 2012.

[12] Franke, U.; and Brynielsson, J., "Cyber situational awareness--a systematic review of the literature," Comput. Secur., vol. 46, pp. 18–31, 2014.

[13] Franke, U.; Johnson, P.; and König, J., "An architecture framework for enterprise IT service availability analysis," Softw. Syst. Model., vol. 13, no. 4, pp. 1417–1445, 2014.

[14] Holm, H.; Buschle, M.; Lagerström, R.; and Ekstedt, M., "Automatic data collection for enterprise architecture models," Softw. Syst. Model., pp. 825–841, 2012.

[15] Holm, H.; Shahzad, K.; Buschle, M.; and Ekstedt, M., "P2CySeMoL: Predictive, Probabilistic Cyber Security Modeling Language," Dependable Secur. Comput. IEEE Trans., vol. PP, no. 99, p. 1, 2014.

[16] Johnson, P.; Ullberg, J.; Buschle, M.; Franke, U.; and Shahzad, K, "An architecture modeling framework for probabilistic prediction," Inf. Syst. E-bus. Manag., vol. 12, no. 4, pp. 595–622, 2014.

[17] KTH ICS, "The Multi-Attribute Prediction (MAP) class diagram,"

2014. [Online]. Available: http://www.kth.se/en/ees/omskolan/organisation/avdelningar/ics/research/sa/p/the-multi-attribute-prediction-map-class-diagram-1.387306. [Accessed: 01-May-2014].

[18] KTH ICS, "EAAT," 2014. [Online]. Available: www.ics.kth.se/EAAT. [Accessed: 01-May-2014].

[19] KTH, "VIKING." [Online]. Available: https://www.kth.se/en/ees/omskolan/organisation/avdelningar/ics/research/cc/proj/v/viking-1.407871. [Accessed: 01-Jan-2015].

[20] Lagerström, R.; Franke, U.; Johnson, P.; and Ullberg, J., "A method for creating enterprise architecture metamodels--applied to systems modifiability analysis," Int. J. Comput. Sci. Appl., vol. 6, no. 5, pp. 89–120, 2009.

[21] Lagerström, R.; Johnson, P.; and Höök, D., "Architecture analysis of enterprise systems modifiability--models, analysis, and validation," J. Syst. Softw., vol. 83, no. 8, pp. 1387–1403, 2010.

[22] Liggins, M.; Hall, D.; and Llinas, J., Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition. CRC Press, 2008.

[23] Moser, C.; Junginger, S.; Brückmann, M.; and Schöne, K.-M., "Some Process Patterns for Enterprise Architecture Management," Softw. Eng. 2009 – Work., pp. 19–30, 2009.

[24] Närman, P.; Franke, U.; König, J.; Buschle, M.; and Ekstedt, M., "Enterprise architecture availability analysis using fault trees and stakeholder interviews," Enterp. Inf. Syst., vol. 8, no. 1, pp. 1–25, 2014.

[25] OMG, "UML Resource Page." [Online]. Available: http://www.uml.org/. [Accessed: 02-Feb-2015].

[26] OMG, "OCL," 2012. [Online]. Available: http://www.omg.org/spec/OCL/2.3.1/. [Accessed: 01-May-2014].

[27] Rahm, E.; and Do, H. "Data cleaning: Problems and current approaches," IEEE Data Eng. Bull., vol. 23, pp. 3–13, 2000.

[28] Simonsson, M.; Lagerström, R.; and Johnson, P. , "A Bayesian network for IT governance performance prediction," in Proceedings of the 10th international conference on Electronic commerce, 2008, p. 1.

[29] Sommestad, T.; Ekstedt, M.; and Holm, H., "The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures," Syst. Journal, IEEE, vol. 7, no. 3, pp. 363–373, 2013.

[30] Sommestad, T.; Holm, H.; and Ekstedt, M., "Effort estimates for vulnerability discovery projects," in Proc. of 45th Hawaii International Conference on System Sciences, 2011.

[31] Steinberg, A. N., "Data fusion system engineering," IEEE Aerosp. Electron. Syst. Mag., vol. 16, pp. 7–14, 2001.

[32] Steinberg, A.N.; and Bowman, C.L., "Chapter 2: Revisions to the JDL Data Fusion Model," Multisens. Data Fusion, vol. 3719, no. April, pp. 430–441, 2001.

[33] Steinberg, A.N.; and Rogova, G., "Situation and context in data fusion and natural language understanding," Proc. 11th Int. Conf. Inf. Fusion, FUSION 2008, 2008.

[34] Talend, "Talend Open Studio." [Online]. Available: http://www.talend.com/products/talend-open-studio. [Accessed: 02-Feb-2015].

[35] Tenable, "Nessus," 2015. [Online]. Available: http://www.tenable.com/products/nessus-vulnerability-scanner. [Accessed: 02-Feb-2015].

[36] The PostgreSQL Global Development Group, "PostgreSQL." [Online]. Available: http://www.postgresql.org/. [Accessed: 03-Mar-2015].

[37] Ullberg, J.; Lagerstrom, R.; and Johnson, P, "A framework for service interoperability analysis using enterprise architecture models," in Services Computing, 2008. SCC'08. IEEE International Conference on, 2008, vol. 2, pp. 99–107.

[38] W3C, "XSL Transformations (XSLT) Version 2.0." [Online]. Available: http://www.w3.org/TR/xslt20/. [Accessed: 02-Feb-2015].

[39] Välja, M.; Lagerström, R.; Ekstedt, M.; and Korman, M., "A Requirements Based Approach for Automating Enterprise IT Architecture Modeling Using Multiple Data Sources," Enterp. Distrib. Object Comput. Work. (EDOCW), 2015 IEEE 19th Int., pp. 79–87, 2015.