Visual and Ontological Modeling and Analysis Support for Extended Enterprise Models

Sagar Sunkle(✉) and Hemant Rathod

Tata Research Development and Design Center, Tata Consultancy Services,
54B, Industrial Estate, Hadapsar, Pune 411013, India
{sagar.sunkle,hemant.rathod}@tcs.com
http://www.tcs.com

Abstract. To remain competitive in dynamic environment, enterprises need to make effective and efficient decisions in response to changes. Good modeling tools and analysis support is a necessity in this regard. Such modeling and analysis tools should to be able to visually model and programmatically analyze several descriptive and prescriptive modeling languages in concert. We recount our experience with visual modeling editor and ontological representation for both descriptive and prescriptive models for enterprise decision making. Starting with purposive modeling tools, we shifted to integrated modeling environment where all relevant models of enterprise coexist and are analyzed together. Our ongoing research suggests that apart from integrated modeling environment, scalable modeling facilities for better interaction between modelers and domain experts are also necessary to make modeling and analysis of enterprise models more streamlined.

Keywords: Enterprise modeling · Intentional models · Visual modeling · Enterprise analysis · ArchiMate · Adex

1 Introduction

Modern enterprises face external change drivers such as evolving market conditions, pressure from competitors' innovation, technology obsolescence and advance, dynamic supply chains and complying with increasingly strict regulations. While external change drivers often demand enterprise transformation [1], internal policies within enterprises tend to be targeted at improvement of enterprise's business as usual (BAU) situation [2]. To remain competitive, enterprises need to respond to changes in an efficient and effective manner. Current state of the art and practice in enterprises rely extensively on expert knowledge with much of the artifacts being document-oriented [1]. We take the stance that responding to changes requires a model-based treatment of four tasks- creating the AS-IS enterprise architecture (EA), coming up with possible TO-BE EAs, devising a way to evaluate TO-BE EAs based on some criteria, and enabling the operationalization of the desired TO-BE EA.

© Springer International Publishing Switzerland 2015
S. Nurcan and E. Pimenidis (Eds.): CAiSE Forum 2014, LNBIP 204, pp. 233–249, 2015.
DOI: 10.1007/978-3-319-19270-3_15

Using purpose-specific, machine-processable models to capture AS-IS and TO-BE EAs, it becomes possible to lessen the reliance on experts by constructing meaningful analyses on such models. The ability to model enterprise change contexts along with rationale is required to be able to relate reasons behind design decisions with on-ground actions of enterprise in both transformation and BAU improvement scenarios. For this we chose ArchiMate [3] and i* [4] as respective modeling languages [5] and Archi [6] and OpenOME [7] as respective modeling tools.

Since the EA and intentional models existed in separate tools, it was not possible to construct combined EA- and intentional modeling-specific analyses. This created problems in terms of traceability between intentional models and elements of TO-BE EA, as it was not preserved in this approach [5]. We had to adopt an ad-hoc process in enacting desired TO-BE EA model presuming that concerns expressed in intentional models are represented completely and consistently in TO-BE EA model.

In accordance with these pointers, we extended Archi to enable integrated visual modeling of EA models as descriptive models and intentional and motivation models (called IM models henceforth) as prescriptive models. We also extended our EA ontology [1], originally containing only EA modeling elements, similarly to carry out various analyses required in terms of prescriptive courses of actions from AS-IS to TO-BE EAs. Our ongoing model building effort with real world case studies suggests that integrated visual modeling support simplifies and streamlines modeling process. At the same time, integrated ontological modeling support enables expressing requisite analyses with ease.

Yet, we perceive the need to be able to carry out both modeling and analysis of EA and IM models using single modeling framework that is capable of providing metamodeling, visualization, and programming support. In practical implementation of EA and IM models, there would be other needs in terms of scalability as well as support for a method for EA and IM modeling and analysis. For this, we propose to use our own reflexive (meta-) modeling framework [8,9] which includes all the above necessary components as well as scalability features.

This paper recounts our experience from our experiments with explicit intentions in EA change response from using separate visual modeling environments to carrying out the visual modeling in integrated modeling environment [10]. Sections 2 and 3 discuss the requisite elements for modeling change response in enterprise context and transition to integrated visual and ontological modeling environment respectively. In Sect. 4, we outline the proposal for knowledge transfer to our modeling environment which has proven scalability features. Section 5 discusses related work and several observations. Section 6 concludes the paper.

2 Modeling Enterprises with Explicit Intentions

With more than 15 years of experience in delivering 70+ large business-critical enterprise applications, we know that the cost of incorrect decisions is prohibitively high in building and evolving these applications, especially in the face

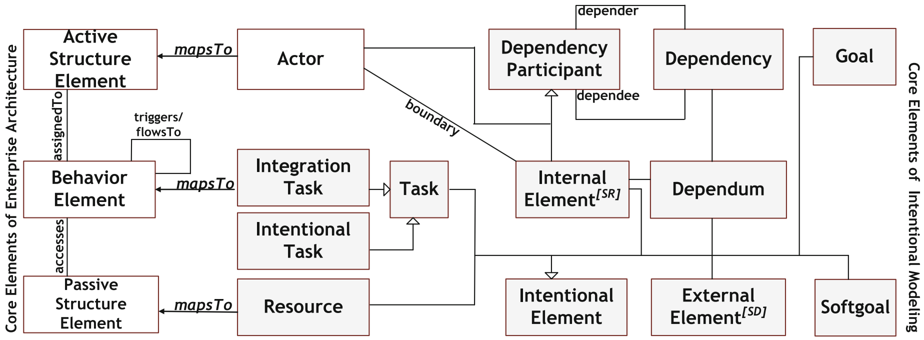

Fig. 1. Mapping EA and intentional modeling elements [5]

of imminent changes. In this regard, we investigated an approach for modeling EA with explicit intentions of enterprise actors. Such a treatment would enable modeling and analyzing strategic alternatives available to enterprise in response to change. We chose ArchiMate for its economy of core concepts and coverage of concepts pertaining to various aspects of enterprise [5]. We chose i* for representing strategic alternatives because of its actor-orientation and availability of evaluation algorithms for ranking strategic alternatives.

We represented AS-IS enterprise architecture (EA) models using Archi [6]. Intentional models were treated as models of the problems an enterprise is trying to solve [5]. These were modeled using OpenOME [7]. We came up with a metamodel mapping between ArchiMate metamodel and intentional metamodel, enabling us to derive intentional models from AS-IS EA models. We represented strategic alternatives in given change situation in OpenOME and evaluated them with label propagation algorithm [11]. Only a single alternative from amongst the optimum alternatives was then chosen to be realized on top of AS-IS EA [5]. The next section describes the elements necessary to model and analyse enterprise rationale using the mapping between EA and intentional modeling elements shown in Fig. 1.

2.1 Requisite Elements for Modeling and Analyzing Explicit Rationale

The ArchiMate generic metamodel defines active structure elements (ASEs) as the entities that are capable of performing behavior. These are assigned to behavior elements (BEs), which indicate units of activity. The passive structure elements (PSEs) are the objects on which behavior is performed [3].

The i* metamodel from [12] considers actors performing tasks as means to ends that are captured as (soft) goals [13]. Resources may be used or created by actor while performing tasks. Actors may depend on each other to perform a task and/or to use or create a resource to achieve a goal or soft goal. Two kinds of models are used in i* namely, strategic dependency (SD) and strategic rationale

(SR) models, to capture dependencies between actors and to model the intentions of actors in performing their appointed tasks respectively. SR models describe reasoning that actors employ in determining the merit in organizing their tasks one way or the other. SD models describe an enterprise in terms of dependencies that enterprise actors have on each other in accomplishing their work.

We mapped the active structure entities (ASEs) such as business actors, application components, hardware and system software, as well as interfaces to actors in i*. The behavioral entities (BEs) such as business processes, and business, application, and infrastructure services were mapped to tasks in i*. The passive structure entities (PSEs) were mapped to resources in i*, which are essentially informational or physical entities used or created by actors. Via this mapping, it was possible for us to model facts of enterprise summarized by *ASEs use or create PSEs while performing BEs as means to ends that are goal(s) or soft goal(s).*

2.2 Using Explicit Rationale

We used this metamodel mapping in a case study in which we re-imagined our Model-driven Engineering-based software development unit as an enterprise and presented two distinct stages in its evolution as current and future states in retrospect [5]. The EA models of this enterprise were developed using Archi [6] and the intentional models were developed using OpenOME [7]. This consisted of the following steps:

1. Obtain/create AS-IS models of the enterprise using Archi.
2. Using these, create intentional models devoid of goals via metamodel mapping using OpenOME.
3. Represent the problems in the change context in terms of goals to be achieved.
4. Model new tasks for original actors and new actors with tasks that might be necessary for achievement of goals modeled in earlier step.
5. Model the dependencies between actors.

The process that an enterprise would follow to go from AS-IS EA to TO-BE EA is shown in Fig. 2. Using the metamodel mapping it was possible for us to represent ASEs, PSEs, and BEs in EA models modeled in Archi in terms of actors, resources, and tasks in intentional models in OpenOME. Using OpenOME's in built alternative evaluation mechanism based on satisfaction label propagation [11], the optimum alternative would be decided.

This alternative existed in the intentional model. The new set of actors and their dependencies would be then transferred back on top of AS-IS EA model via metamodel mapping, thus giving a specific TO-BE model. This TO-BE model would dictate what needs to be operationalized on ground to reach desired response to change under consideration.

This arrangement worked as far as a specific TO-BE model was to be built. As we applied the same process to create considerably larger models [14], we found that:

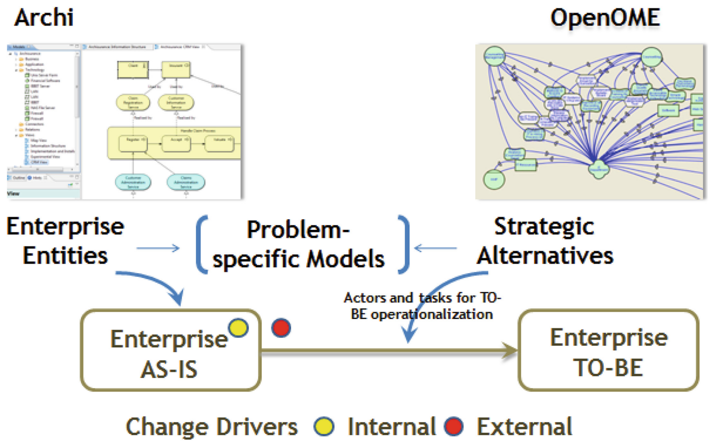

Fig. 2. Using explicit rationale for transition to desired TO-BE EA [5]

- For really large enterprise models containing thousands of entities, keeping the EA and intentional models in sync in two different modeling tools became nearly impossible as models started to grow in size.
- In our original approach, the transformation of intentional models from EA models and the other way round was manual because of the independent tools.
- This also meant that only a single TO-BE EA model could be obtained at a time. Ideally, it should be possible to derive any TO-BE EA model corresponding to selected optimum strategic alternative.

In short, we needed a way to be able to visually model both EA and intentional concerns together. Our ongoing efforts in this regard are discussed in the next section.

3 Modeling EA and IM with Integrated Visual Modeling Support

As explained at the end of last section, we needed both visual modeling and analysis support for combined EA and intentional models. Furthermore, the process of capturing AS-IS had to be more guided as otherwise domain experts were found to be at loss as to what it is that modelers wanted to model in a particular change context. Finally, the analysis had to make a way for preserving the information about specific alternatives in the TO-BE EA models.

3.1 Adding Motivation Elements to Intentional Models

In spite of the fact that ArchiMate motivation extensions provide goal related elements already [15], we chose intentional modeling for capturing goals for the following reasons:

1. Actors should own the tasks that are used as means to achieve certain ends. Intentional models take the actor/agent-oriented view in SR models whereas ArchiMate's treatment in motivation extensions is more generic as well as implicit via *associated with* links.
2. Dependencies between actors are captured in intentional models in SD models; ArchiMate motivation extensions do not provide any specialized elements for capturing dependencies.
3. Qualitative aspects of a solution to problem in enterprise change context are not explicitly captured in motivation extensions as opposed to soft goals and their semantics in intentional modeling.
4. Finally, motivation extensions do not provide any evaluation mechanism for strategic alternatives similar to satisfaction label propagation in intentional modeling [11].

As we started modeling EA and intentional models of various problems in real world case studies, we found that domain experts were more at home with elements like *internal* and *external drivers (motivations)*, *stakeholders, assessment,* and so on, although they agreed that sharp goal definitions with qualitative aspects were also used in change scenarios. Drivers, both within an enterprise and from the enterprise's environment, influence rest of the IM elements. Generally, a stakeholder becomes interested in assessment of a driver and it is this assessment that leads to formulation of a goal. From thereon, intentional modeling begins in terms of actor who is responsible for achieving the goal and actions that need to be taken by that actor, in most cases depending on other actors. Reader is redirected to some of our case studies presented in [2,16] for further details.

We needed to amend the intentional metamodel with motivation related elements for better articulation in interaction between modelers and domain experts. Also, earlier approach consisted of including additions and modifications to EA elements of AS-IS model to represent the TO-BE model. Instead of this ad hoc process, a model-based process would be more helpful where for a given strategic alternative, the corresponding TO-BE model could be obtained without manual adjustment to the AS-IS EA model.

Figure 3 shows the extended enterprise metamodel with motivation modeling elements. This mapping enabled us to specify that *key stakeholders' assessments of external and/or internal drivers of enterprise lead to goals and soft-goals that active structure entities are motivated to achieve by performing behavior (entities) and using or creating passive structure entities.*

To streamline this kind of articulation of problems in enterprise change context as well as to overcome limitations of independent modeling environments, we created integrated visual modeling support. We chose Archi as it already supports modeling the business, application, and infrastructure layers of ArchiMate [6] and we would have to add only the IM elements to it. The next section describes how we created the integrated visual modeling environment.

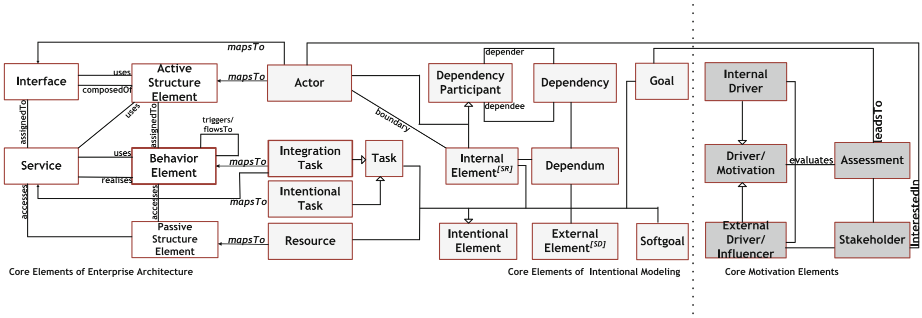

Fig. 3. Extending enterprise metamodel in Fig. 2 with motivation elements

3.2 Enabling Integrated Visual Modeling Environment

Archi is based on Eclipse Modeling Framework (EMF). The process of adding IM elements and relations to base EA metamodel in Archi consisted of adding IM element- and relation-specific classes and then specifying permitted relations between specific elements.

After this, visualization aspects of elements and relations need to be defined. By defining all the elements required by EA and IM models and sets of allowed relations between various elements, EA and IM models can be drawn as shown by ③ in Fig. 4. The models panel in Fig. 4 enlists AS-IS EA models as well as IM models of these problems we refer to as global views. ④ shows such a global view of products and services rationalization problem. Current version of Archi

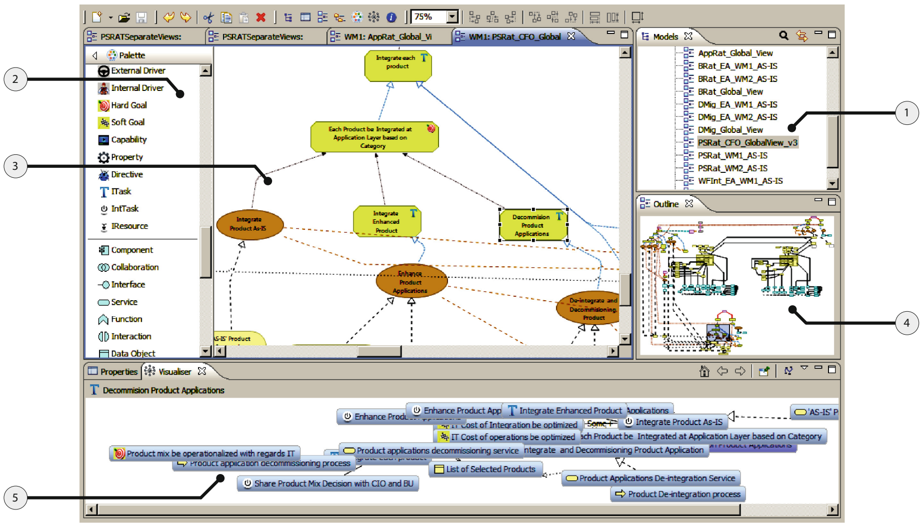

Fig. 4. Extended Archi for EA and IM modeling

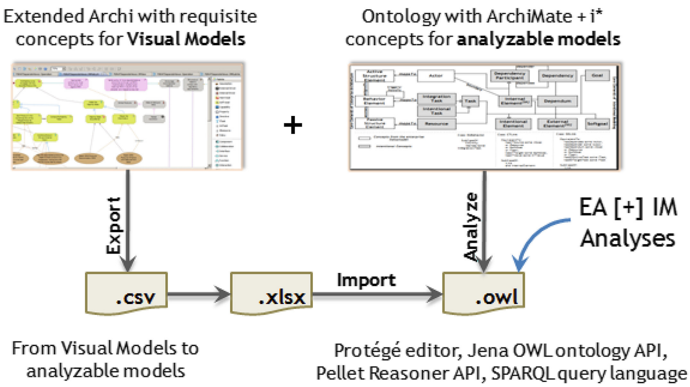

Fig. 5. Architecture of integrated EA and IM modeling environment

supports visualization of a selected model element in terms of all other elements that it is related to as shown by ⑤ in Fig. 4.

In Fig. 4, ① shows list of integrated models enterprise under consideration with the facility to model both EA and IM elements. The extended Archi elements based on the extended enterprise metamodel in Fig. 3 are shown by ② in Fig. 4. With the core metamodel of Archi extended to metamodel in Fig. 3, it became possible to visually model both EA and IM elements together.

The architecture of integrated tooling is shown in Fig. 5. We were already using ontological representation of EA models for easier specification of EA-specific analyses. While extended Archi enabled visual modeling, constructing analyses in EMF would require considerable boilerplate coding which could be easily done away by extending EA ontology with IM elements and then specifying various analyses using ontology APIs as shown in Fig. 5. Archi enables export to CSV files which retain EA element type and name of both source and target nodes along with relation and documentation if any. The next section describes how the ontological representation is leveraged for EA- and IM-specific as well as combined analyses.

3.3 Extended Enterprise Ontology for Purposive Analysis

We presented an ontological representation that captures ArchiMate's core metamodel as well as layer specific metamodels. This is shown in Fig. 6. The relations between generic elements reflect in each of business, application, and infrastructure layers. This is shown in the middle of Fig. 6 with two example elements each in business and application layers. An active structure element is assignedTo a behavior element. In the business layer, a BusinessRole is assignedTo a BusinessProcess. A behavior element accesses a passive structure element. In the application layer, an ApplicationFunction accesses an ApplicationDataObject. This representation was versatile enough for conducting change impact and landscape mapping analyses. For a more complete elaboration of these analyses, reader is requested to refer to [1].

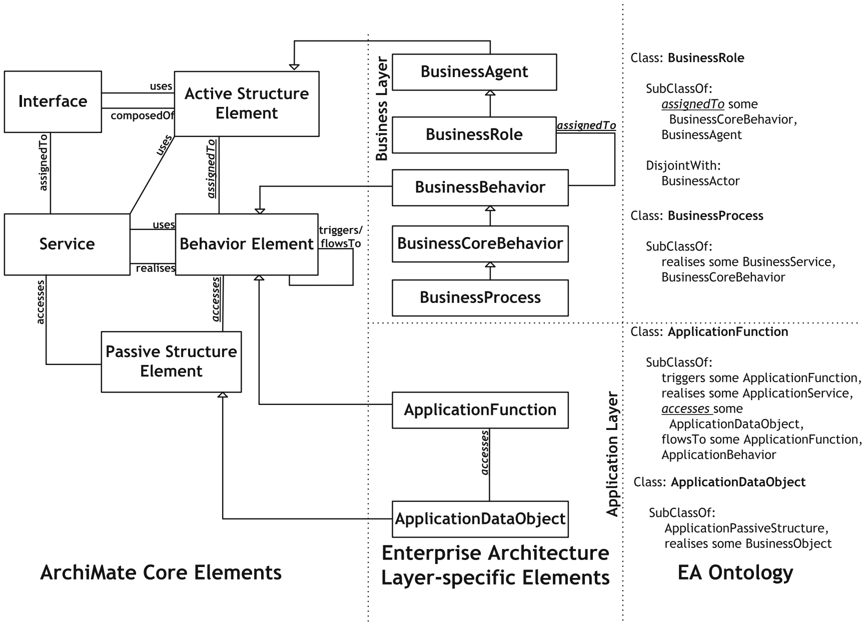

Fig. 6. EA-based ontology [1]

We extended the EA ontology presented in [1], with intentional elements as shown in Fig. 7. We found that all of the relations in intentional model namely, means-ends (MELink), task decomposition (TDLink), contribution (CTLink), and strategic dependency (SDLink) relations, benefit from being represented as *reified* relations [17]. For instance, a contribution link indicates not only which element contributes to a soft goal but also what that contribution is. This ontology also includes motivation elements shown in Fig. 3.

The model exported from Archi is easily read into ontology model by first constructing the dictionary, leveraging the type information of instances and then constructing the data in terms of relations.

We import an important convention introduced in the ArchiMate motivation extensions to connect requirements of goals with behavior of active structure entities [3, ArchiMateSpecification 10.2.5]. This convention is captured in the ontology in the definition of EABehavior class which realizes some IntegrationITask element from IM models. Integration tasks essentially represent leaf tasks in SR models which are not dependent on any other elements. These are treated as *primitively workable* elements [4] and used in defining operationalization models of TO-BE EA via connecting elements which are instances of sub-classes of EABehavior class.

3.4 Conducting Combined EA and IM Model Analysis

We have implemented the bottom-up label propagation algorithm in [11] to compute satisfaction level of the root goal. The ontological representation easily

```
Class: EABehavior
  SubClassOf:
    EAEntity,
    realises some
IntegrationITask
```

```
Class: MELink
  EquivalentTo:
    hasMEGoal some IGoal,
    hasMETask some ITask

  SubClassOf:
    ILink
    and InternalElement
```

```
Class: TDLink
  EquivalentTo:
    hasTDSourceTask some
ITask,
    hasTDTarget some (IGoal
    or IResource
    or ISoftGoal
    or ITask)

  SubClassOf:
    ILink
    and InternalElement
```

```
Class: CTLink
  EquivalentTo:
    hasCTSource some (IGoal
    or IResource
    or ISoftGoal
    or ITask),
    hasCTTarget some ISoftGoal,
    hasCTValue some ICTValue

  SubClassOf:
    ILink
    and InternalElement
```

```
Class: SDLink
  EquivalentTo:
    hasDependee some IActor,
    hasDepender some IActor,
    hasDependum some (IGoal
    or IResource
    or ISoftGoal
    or ITask),
    hasSDSourceTask some ITask,
    hasSDTargetTask some ITask

  SubClassOf:
    ILink
```

Fig. 7. Extending EA ontology with IM elements and relations [2]

enables implementing the label propagation as well as computation of specific routines in terms of ontology APIs. The concept of actor's routine is central in computing optimum alternatives. It is the set of tasks an actor needs to carry out to achieve a goal which may also include set of tasks of other actors via dependency links. Both label propagation and actors' routine computation is carried out by recursively traversing down the intentional graph starting with the root goal and using queries to traverse over means-ends, task decomposition, contribution, and strategic dependency links.

Leaf tasks in IM models are related to EA elements that will operationalize them from TO-BE EA perspective. It is possible that in some cases some of the AS-IS elements could be reused, mostly by reference through a relation between newly added element and existing AS-IS element. In order to preserve which elements were added for specific leaf tasks, we tag them in the ontological representation. With this tagging, the EA elements that are added anew in contrast to AS-IS elements are identified. Using this combined analysis, both EA and IM models can be refined such that they capture the reality to the satisfaction of domain experts [10].

3.5 Toward Industry Strength Modeling and Analysis for Enterprises

With ontological representation we have been able to apply various decision making analyses to EA and IM models thus created. Yet, our experience in development of model-driven applications suggests that for practical implementation of EA and IM modeling and analyses, we need scalable tooling [18]. In the next section, we propose to use our proprietary reflexive (meta-) modeling framework called **Adex** which we have successfully used in over 70+ applications in multiple domains for organizations spread across the globe.

4 Scalable Modeling and Analysis for Enterprise Decision Making

Our current tooling for modeling and analysis is based mainly on visual modeling of EA and IM elements enabled by extended Archi and ontological representation to which extended Archi models are exported. We are using this tooling for experimenting with various EA and intentional decision making modeling and analyses. Eventually though, EA and IM modeling and analysis will need to be supported in industry strength tooling with proven scalability characteristics. We propose to implement the EA and IM modeling support showcased so far in proprietary (meta-) modeling framework Adex [8].

4.1 Enabling EA and IM Modeling and Analysis in Adex

The process of creating purpose-specific metamodel in Adex is similar to EMF. Visualization functionality is provided with Adex through *symbol designer* as shown on the right of Fig. 8. Visual aspects of elements and relations can be defined not unlike in EMF-based Archi. The minor difference is that symbols and connectors for elements and relations can be defined separately and then mapped to existing elements and relations, whereas in Archi, specific classes related to defined elements and relations have to be extended in specific ways manually to construct visual representations.

The ontological representation to which Archi models are exported enables flexible analysis implementation with various ontology APIs [1]. A similar support is provided in Adex with *OMGen* language, which is a (meta-) model-aware language [8]. OMGen scripts can be used to perform desired actions such as initialization of instances, value computation and propagation to related instances, performing validations, and triggering external actions. Scripts can interface with external C programs, allowing custom software and third-party utilities to be linked in. Adex's scripting feature can be effectively utilized to integrate modeling activity with analysis building.

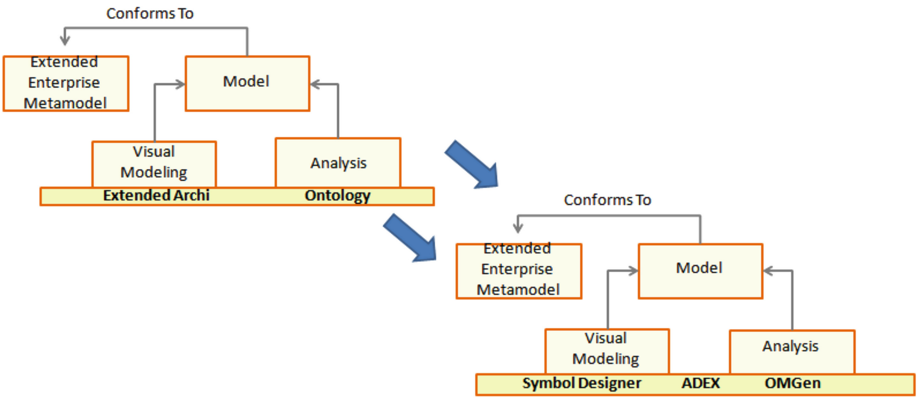

Fig. 8. Shifting modeling and analysis for EA and IM models from extended Archi to Adex

Table 1. Modeling and analysis features for EA and IM models [2]

Modeling and/or Analysis Feature	Archi	Adex
Metamodeling	Yes	Yes
Visual Modeling	Yes	Yes
Scripting for flexible code and report generation	No	OMGen
Domain-specific checks for model consistency	Predefined	Predefined + configurable
Manage work units	No	Packages
Data File/DB support	Stored in XML, export to CSV available	File, in memory, MySQL, Oracle, etc.
Multi-user/role support	No	Yes
Versioning	No	Yes
Diff/Merge	No	Yes

Italic Font- Common features, ***Italic Bold Font-*** Adex-specific features

While SPARQL[1] query language provides SQL-like syntax and functionality that is declarative in nature, OMGen is an imperative language. We may need to provide a declarative wrapper over imperative scripts, as we have found SPARQL like syntax to be quite effective in building EA [1] and intentional analyses.

4.2 Scalability Features for EA and IM Models

As shown in Fig. 8, both visual modeling and analysis can be supported out of the box using Adex. But Adex provides substantially different functionality beyond representation and manipulation of (meta-) models that is a must for highly scalable modeling activity including EA and IM modeling and analyses.

Table 1 distinguishes between such requisite features from the common ones between Adex and (extended) Archi. There are set of features that Adex supports which we have found to be absolutely critical for scalable model-driven development and we think that these features will also be needed to scale EA and IM models of enterprises. We discuss these features in brief here. For detailed explanations, reader is directed to [8,18]. Note that the lack of scripting over Archi models is balanced by ontological representation in our ongoing approach. Ontological representation can be used also to include domain-specific consistency checks. In Adex, this can be done with OMGen.

While Adex's metamodel is similar to OMG's metaobject facility, Adex provides much richer set of constructs that enable model consistency checks, work units partitioning, database, multi-user/role support, versioning and diff-merge abilities [8].

Adex uses *components* for partitioning given models. Versioning of models is carried out at component level. *Configurations* are provided as containers for assembling different versions of components. A configuration is what represents the notion of a *work unit*. Inter-component associations (between objects

[1] http://www.w3.org/TR/rdf-sparql-query/.

belonging to different component versions) provide a mechanism to establish and enforce compatibility semantics between two component versions.

When a component is developed concurrently by independent teams, diff-merge operations can be used to do so in a non-interfering way at configuration, component, or object level. Both metamodels and models are stored in separate repositories. Together these features have been and are being used in several active assignments with model size often rising to many gigabytes and several teams with differing roles across the globe using these models.

We believe that in the context of combined EA and IM modeling and analysis, such features will become essential and with Adex we are poised to transfer our ongoing efforts to a unified and scalable modeling and analysis environment.

5 Related Work

Elements for Explicit Intentions, Motivation, and Operationalization. ArchiMate provides essential elements for modeling business, application, and infrastructure layers of an enterprise. Further extensions such as ArchiMate motivation extensions add motivation/goal related elements [15]. As discussed earlier in Sect. 3.1, from strategic analysis point of view, we preferred to use intentional modeling while adding some elements from motivation extensions as well as referring to business motivation model (BMM) for better understanding of motivation elements. We tried to keep new elements less in number as similar studies related to requirements modeling languages such as ARMOR language [19], have indicated that adding several elements from different goal modeling languages is not useful from practitioners' perspective [20].

Our focus in selecting additional elements on top of regular intentional modeling elements was to make the process of modeling EA and intentional aspects of enterprise easier for modelers and domain experts. At the same time, we found it necessary to record assessments of drivers because assessments and assumptions influence the rest of decision making. Our ongoing research with regards directives, policies, and regulations compliance makes use of these concepts and follows BMM's logical progression in reaction to change [21, Fig. 7.2]. With regards specific elements, our internal surveys revealed that domain experts are often exposed to notions of drivers or influencers as well as sharply defined goals (internally we use balanced scorecard for capturing goals). Some of our observations are contrary to other studies such as [20], which found goal and motivation elements like means-ends relationship, distinction between hard and soft goals, and assessment did not make much sense to practitioners. We believe that every enterprise should decide on the set of elements that all the key actors in that enterprise agree on as to what they mean and how useful they are for capturing AS-IS and TO-BE EAs for decision making.

Failure of strategic response to changes is found to be rooted in the lack of strategy operationalization [22]. Still goal modeling approaches for enterprises tend to focus only on strategy making, often referred to as problem of constructing design/solution alternatives, and evaluation of strategic alternatives

[20,23,24]. Operationalization, if at all, is considered from business perspective as in strategy maps [25] or from IT perspective [26]. Also operationalization is either presumed to just happen or it is generally just a set of guidelines. We have borrowed the notion of goal requirements realized by behavior entities in EA from ArchiMate. We treat this as the glue between suggested alternatives and TO-BE EA. In future, we intend to investigate the ways in which TO-BE EA is operationalized via human resources and IT systems. We have presented some initial work in this regards with respect to operationalizing internal policies by persisting suggested alternatives to business processes [2], which is in line with ArchiMate's implementation and migration extensions [3].

Visualization of EA and other Models. An approach is suggested in [27], which enables visualization of purposive analyses upon EA model. This EA modes conforms to metamodel that incorporates elements from ArchiMate, BMM, and Business Process Modeling Notation, etc. Their modeling environment visualizes purposive analyses like business-IT alignment in terms of paths between motivation and IT elements using visualization techniques like sunburst. In contrast our work focuses on using visual environment for modeling alone, whereas analyses are always conducted using ontological representation.

For capturing enterprise transformation to TO-BE states, graphical viewpoints on TO-BE states are suggested in [28,29], whereas [30] proposes to visualize migration roadmap. As shown earlier, our approach so far has been to use visual modeling for creating the models and then using ontological representation for coming up with TO-BE states as well as paths to TO-BE states. The models used in these are also EA models without other purposive models like intentional models.

Analysis of EA and Other Models. Most research in EA analysis focuses on analysis of EA models themselves, e.g., change impact analysis of EA models [31], quantitative analysis of EA models [32], and analysis of various non-functional properties in EA while incorporating treatment of uncertainty in EA models [33–35]. In contrast, we treat EA models as descriptive models and use intentional models [5] and system dynamic models [36] as prescriptive models upon which to base decision making. The modeling and analysis need to be supported by tooling environment that supports visual modeling of purposive modeling languages and also programmable analysis.

6 Conclusion

Our ongoing explorations with visual modeling support for extended enterprise models suggests that integrated visual modeling streamlines the enterprise modeling activity while integrated ontology modeling support enables implementing requisite analyses. Visual models are imported into ontology because of which, models and analysis results remain in sync. While visual modeling support explained in the paper is working as desired, we suspect that when many modelers are simultaneously modeling various problem-specific IM and enterprise models in multiple interactions with domain experts, a more robust distributed enterprise and IM modeling environment will be necessary. We have outlined how current support for enterprise and IM modeling can be shifted from open source Archi to our own (meta-) modeling framework Adex. Our long term goal in this regard is to make enterprise and purposive modeling and analysis as simple and effective as it is when it comes to code generation.

References

1. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Analyzing enterprise models using enterprise architecture-based ontology. In: Moreira, A., Schätz, B., Gray, J., Vallecillo, A., Clarke, P. (eds.) MODELS 2013. LNCS, vol. 8107, pp. 622–638. Springer, Heidelberg (2013)
2. Sunkle, S., Kholkar, D., Rathod, H., Kulkarni, V.:Incorporating directives into enterprise TO-BE architecture. In: [37], pp. 57–66
3. Iacob, M., Jonkers, D.H., Lankhorst, M., Proper, E., Quartel, D.D.: Archimate 2.0 Specification. The Open Group, Van Haren Publishing, Reading (2012)
4. Yu, E.S.K.: Modelling strategic relationships for process reengineering. Ph.D. thesis, Toronto, Ontario, Canada (1996). UMI Order No. GAXNN-02887 (Canadian dissertation)
5. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Intentional modeling for problem solving in enterprise architecture. In: Hammoudi, S., Maciaszek, L.A., Cordeiro, J., Dietz, J.L.G. (eds.) ICEIS (3), pp. 267–274. SciTePress (2013)
6. Institute, E.C.: Archi Developer Resources. http://archi.cetis.ac.uk/developer.html. Accessed 16 Mar 2014
7. Horkoff, J., Yu, Y., Yu, E.S.K.: OpenOME: an open-source goal and agent-oriented model drawing and analysis tool. In: de Castro, J.B., Franch, X., Mylopoulos, J., Yu, E.S.K. (eds.) iStar. CEUR Workshop Proceedings, vol. 766, pp. 154–156. CEUR-WS.org (2011)
8. Reddy, S., Mulani, J., Bahulkar, A.: Adex – a meta modeling framework for repository-centric systems building. In: 10th International Conference on Management of Data (COMAD) (2000)
9. Kulkarni, V., Reddy, S.: Integrating aspects with model driven software development. In: Al-Ani, B., Arabnia, H.R., Mun, Y. (eds.) Software Engineering Research and Practice, pp. 186–197. CSREA Press (2003)
10. Sunkle, S., Rathod, H.: Visual modeling editor and ontology API-based analysis for decision making in enterprises - experience and way ahead. In: [37], pp. 182–190
11. Horkoff, J., Yu, E.S.K.: Evaluating goal achievement in enterprise modeling – an interactive procedure and experiences. In: Long, M. (ed.) World Congress on Medical Physics and Biomedical Engineering May 26-31, 2012 Beijing, IFMBE Proceedings, vol. 39, pp. 145–160. Springer, Heidelberg (2013)
12. Ayala, C.P., Cares, C., Carvallo, J.P., Grau, G., Haya, M., Salazar, G., Franch, X., Mayol, E., Quer, C.: A comparative analysis of i*-based agent-oriented modeling languages. In: Chu, W.C., Juzgado, N.J., Wong, W.E. (eds.) SEKE, pp. 43–50 (2005)
13. López, L., Franch, X., Marco, J.: Making explicit some implicit *i** language decisions. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 62–77. Springer, Heidelberg (2011)

14. Sunkle, S., Kulkarni, V., Rathod, H.: (Multi-) modeling enterprises for better decisions. In: Combemale, B., DeAntoni, J., France, R.B., Barn, B., Clark, T., Frank, U., Kulkarni, V., Turk, D. (eds.) GEMOC+AMINO@MoDELS. CEUR Workshop Proceedings, vol. 1102, pp. 69–78. CEUR-WS.org (2013)
15. Quartel, D., Engelsman, W., Jonkers, H.: ArchiMate ® Extension for Modeling and Managing Motivation, Principles and Requirements in TOGAFT™, October 2010
16. Sunkle, S., Rathod, H., Kulkarni, V.: Practical goal modeling for enterprise change context: a problem statement. In: Indulska, M., Purao, S. (eds.) ER Workshops 2014. LNCS, vol. 8823, pp. 139–144. Springer, Heidelberg (2014)
17. Group, W.W.: Defining N-ary Relations on the Semantic Web. http://www.w3.org/TR/swbp-n-aryRelations/. W3C Working Group Note, 12 April 2006. Accessed 16 Mar 2014
18. Kulkarni, V., Reddy, S., Rajbhoj, A.: Scaling up model driven engineering – experience and lessons learnt. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010, Part II. LNCS, vol. 6395, pp. 331–345. Springer, Heidelberg (2010)
19. Quartel, D.A.C., Engelsman, W., Jonkers, H., van Sinderen, M.: A goal-oriented requirements modelling language for enterprise architecture. In: EDOC, pp. 3–13. IEEE Computer Society (2009)
20. Engelsman, W., Wieringa, R.: Goal-oriented requirements engineering and enterprise architecture: two case studies and some lessons learned. In: Regnell, B., Damian, D. (eds.) REFSQ 2011. LNCS, vol. 7195, pp. 306–320. Springer, Heidelberg (2012)
21. OMG: Business Motivation Model - Version 1.2., May 2014
22. Ross, J.W., Weill, P., Robertson, D.: Enterprise Architecture As Strategy: Creating a Foundation for Business Execution. Harvard Business School Press, Boston (2006)
23. Yu, E.S.K., Strohmaier, M., Deng, X.: Exploring intentional modeling and analysis for enterprise architecture. In: Tenth IEEE International Enterprise Distributed Object Computing Conference (EDOC) Workshops, p. 32 (2006)
24. Bryl, V., Giorgini, P., Mylopoulos, J.: Supporting requirements analysis in tropos: a planning-based approach. In: Ghose, A., Governatori, G., Sadananda, R. (eds.) PRIMA 2007. LNCS, vol. 5044, pp. 243–254. Springer, Heidelberg (2009)
25. Giannoulis, C., Zdravkovic, J., Petit, M.: Model-driven strategic awareness: from a unified business strategy meta-model (UBSMM) to enterprise architecture. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) EMMSAD 2012 and BPMDS 2012. LNBIP, vol. 113, pp. 255–269. Springer, Heidelberg (2012)
26. Frank, U.: Multi-perspective enterprise modelling as a foundation of method engineering and self-referential enterprise systems. In: Hammoudi, S., Maciaszek, L.A., Cordeiro, J., Dietz, J.L.G. (eds.) ICEIS (1), pp. IS-9. SciTePress (2013)
27. Naranjo, D., Sánchez, M.E., Villalobos, J.: Connecting the dots: examining visualization techniques for enterprise architecture model analysis. In: Grabis, J., Kirikova, M., Zdravkovic, J., Stirna, J. (eds.) Short Paper Proceedings of the 6th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2013), Riga, Latvia, 6–7 November 2013. CEUR Workshop Proceedings, vol. 1023, pp. 29–38. CEUR-WS.org (2013)
28. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Heidelberg (2005)
29. Niemann, K.D.: From Enterprise Architecture to IT Governance. Elements of Effective IT Management, 1st edn. Vieweg+Teubner, Wiesbaden (2006)

30. Kim, H., Lee, R.Y. (eds.): 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, SNPD 2009, in conjunction with 3rd International Workshop on e-Activity, IWEA 2009, 1st International Workshop on Enterprise Architecture Challenges and Responses, WEACR 2009, Catholic University of Daegu, Daegu, Korea, 27–29 May 2009. IEEE Computer Society (2009)
31. de Boer, F.S., Bonsangue, M.M., Groenewegen, L., Stam, A., Stevens, S., van der Torre, L.W.N.: Change impact analysis of enterprise architectures. In: Zhang, D., Khoshgoftaar, T.M., Shyu, M.L. (eds.) IRI, pp. 177–181. IEEE Systems, Man, and Cybernetics Society (2005)
32. Iacob, M., Jonkers, H.: Quantitative analysis of enterprise architectures. Proceedings of the First International Conference on Interoperability of Enterprise Software and Applications (INTEROP-ESA), February 2005
33. Buschle, M., Ullberg, J., Franke, U., Lagerström, R., Sommestad, T.: A tool for enterprise architecture analysis using the PRM formalism. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 108–121. Springer, Heidelberg (2011)
34. Ekstedt, M., Franke, U., Johnson, P., Lagerström, R., Sommestad, T., Ullberg, J., Buschle, M.: A tool for enterprise architecture analysis of maintainability. In: Winter, A., Ferenc, R., Knodel, J. (eds.) CSMR, pp. 327–328. IEEE (2009)
35. Franke, U., Johnson, P., Ericsson, E., Flores, W.R., Zhu, K.: Enterprise architecture analysis using fault trees and MODAF. In: Proceedings of CAiSE Forum 2009, vol. 453, pp. 61–66, June 2009. ISSN 1613–0073
36. Sunkle, S., Roychoudhury, S., Kulkarni, V.: Using intentional and system dynamics modeling to address WHYs in enterprise architecture. In: Cordeiro, J., Marca, D.A., van Sinderen, M. (eds.) ICSOFT, pp. 24–31. SciTePress (2013)
37. Grossmann, G., Hallé, S., Karastoyanova, D., Reichert, M., Rinderle-Ma, S. (eds.): 18th IEEE International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOC Workshops 2014, Ulm, Germany, 1–2 September 2014. IEEE (2014)