# Towards a More Integrated EA Planning: Linking Transformation Planning with Evolutionary Change

Stephan Aier[1], Sabine Buckl[2], Bettina Gleichauf[1], Florian Matthes[2],
Christian M. Schweda[2], Robert Winter[1]

[1] Institute of Information Management, University of St. Gallen
Müller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
{stephan.aier | bettina.gleichauf | robert.winter}@unisg.ch

[2] Chair for Software Engineering of Business Information Systems (sebis),
Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany
{sabine.buckl | matthes | christian.m.schweda}@mytum.de

**Abstract:** Enterprises are subject to continuous change driven by top-down planned transformation projects as well as by bottom-up initiatives realizing what is called the evolution of the enterprise. Enterprise architecture (EA) planning presents itself as a means for facilitating and controlling this change. Nevertheless, the methods and models of EA planning developed in recent years either have a strong focus on planned (proactive) transformations or on guided (reactive) evolution. In this paper, we outline an EA planning method that accounts for both types of enterprise change by illustrating the interplay of EA planning, requirements, release, and synchronization management. Specifically we focus on the coordination of design activities modeled as intermeshed closed-loop control systems and on an integrated information model describing EA transformation planning.

**Keywords:** Enterprise architecture planning, methods, models, techniques

## 1 Introduction

Today's enterprises find themselves confronted with ever changing economic, regulatory, and technical environments that they are forced to continuously adapt to [RWR06], [Wa05]. Performing changes that are necessary or help to leverage idle opportunities is a complex task, aggravated by the intricate and highly interwoven architecture of the overall organization. Thereby, architecture or enterprise architecture (EA), respectively, is understood as the "fundamental organization of a system [the enterprise] embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" [IOS07].

A commonly accepted instrument to guide enterprise transformations is EA planning, whose main goal is to enhance and maintain the mutual alignment of business and IT [HV93], [LLO93]. Effectively established, EA planning as part of an EA management (EAM) function provides an organization-wide perspective on change, making the impact of projects on EA elements transparent. While business is commonly seen as the driver of organizational change, projects are the implementers and by their nature a cross-cutting aspect. Changes to an enterprise can be differentiated into *evolution* (incremental change, usually bottom-up driven) and *transformation* (fundamental change, usually top-down driven). Whereas support for the former type of change is typically provided by functional methods of business administration, e.g. human resources, distribution, or marketing, the latter requires a holistic management function to systematically support organizational transformation [Wi08]. Although EA planning is commonly regarded as a core capability of EAM [Pu06], only a few authors propose well-engineered methods and models for EA planning. Especially, the need to integrate EA planning with existing management functions like enterprise-wide requirements management or project portfolio management to derive consistent transformation plans as well as synchronization management to implement changes is not considered in existing approaches. Therefore, this paper contributes to the field of EA planning by answering the following research questions:

- How can top-down driven, planned transformation and bottom-up driven guided evolution of the enterprise be consistently integrated in EA planning?
- Which methods or conceptualizations from related disciplines (requirements, project portfolio, and synchronization management) are suitable for consistently planning transformation and evolution aspects of enterprise change?

In order to answer the above questions, we follow the *identify problem/motivate – define objectives – design – demonstrate – [evaluate]* process by [Pe07]. Section 2 motivates the need for an integrated EA planning method and model and derives relevant requirements. Prefabrics from the areas of EA planning, change, and requirements management as well as project portfolio, and synchronization management are revisited in Section 3. Our artifact – a method for integrated EA planning – is presented in Section 4. The theoretic discussion on the artifact is complemented in Section 5 by a case study, which serves as a demonstration and initial evaluation [1]. Finally, Section 6 critically reflects our contribution and provides an outlook on further research.

## 2 Deriving requirements for integrated planning

Matthes et al. conducted a survey on the state-of-the-art in EAM tools [Ma08]. Central means of analyzing these tools is a set of nine “scenarios” that describe typical activities of EAM. The scenario descriptions provide core concerns on an abstract level which are further concretized with questions that are to be answered in performing the activity. While initially developed as a set of requirements for analyzing EAM tools, the scenarios also provide a basis for deriving requirements for an integrated EA planning method and underlying model. In short the methodology for scenario elicitation can be summarized in five steps as follows [Ma08]:

- Derive initial scenario description from project experiences, scientific literature, and end-user input

[1] Although our demonstration is based on a real industry case, it is not hold a full evaluation. However, this single case provides support for the proposed artifact to justify its discussion in the scientific community by this publication.

- Conduct on-site workshops with the survey's industry partners
- Revise scenario descriptions to on-site feedback provided by reviewers and add or reshape scenarios appropriately
- Conduct remote review round with industry partners of the survey
- Revise scenario descriptions to feedback

Following this methodology, a comprehensive set of scenarios based on existing literature and further leveraging the knowledge of 30 user companies has been compiled. Three of these scenarios are essential for EA change processes, namely *enterprise-wide requirements management, project portfolio management,* and *synchronization management*. In the following the concerns of these scenarios are briefly summarized and the corresponding questions are reformulated as requirements for an integrated EA planning method and underlying model.

Enterprise-wide requirements management is concerned with gathering and documenting the requirements raised at different times and in different parts of an organization. A requirement describes a desired state of an organization's EA or part thereof. Affected EA elements can reside on all EA layers from business to infrastructure. In the management process these requirements are classified in respect to their urgency and are grouped by affected EA elements. Thus, typical requirements of enterprise-wide requirements management read as follows:

**(R1)** Collect and document requirements in a uniform way

**(R2)** Categorize requirements in respect to their urgency ("must" vs. "may")

**(R3)** Link requirements to affected EA elements

**(R4)** Derive project proposals from requirements

Project portfolio management decides which project proposals are executed as projects in the next planning period. The process plans a short term future state of the EA based on various measures and takes a transformation-centric perspective by analyzing which EA elements are changed by which project. By doing so, the management process diagnoses potential for alignment between change activities. Such analyses can uncover that a project targets optimizations to an element which is conversely retired by another project. Project portfolio management raises the following requirements:

**(R5)** Estimate project proposals' costs and economic impact

**(R6)** Analyze and resolve dependencies between *transformation* and *evolution* project proposals targeting the same EA element

**(R7)** Ensure that urgent project proposals are prioritized high and scheduled on short time initiation

**(R8)** Assign and distribute project budgets

Synchronization management establishes a common time-schedule for approved projects respecting their interdependencies. In particular, projects that aim at affecting the same EA elements at the same time but may not be integrated into a single project are rescheduled to minimize project interplay. Synchronization management further is no one-time activity but has to continually track project states such that actions for re-scheduling can be taken early, if necessary. Thus, the following requirements can be derived:

**(R9)** Schedule projects to minimize pertaining EA-induced interdependencies

**(R10)** Control project progress and re-schedule in case of project delays

Reflecting our research questions (see Section 1) requirements (**R1**), (**R3**), (**R4**), (**R6**), and (**R9**) are of interest and are addressed by our method and model.

## 3 Preliminary and related work

The question of engineering enterprise transformation using EA models has been addressed in detail by [AG10b]. However, [AG10b] do not consider evolutionary bottom-up change initiatives that may occur on a local project level. We aim at enhancing the existing method towards an integrated approach that reflects methods and models from enterprise-wide requirements, project portfolio, and synchronization management (**R1**) and thus integrates centrally derived transformation plans as well as locally triggered evolutionary change (**R6**). Figure 1 shows how EA planning depends on other processes.

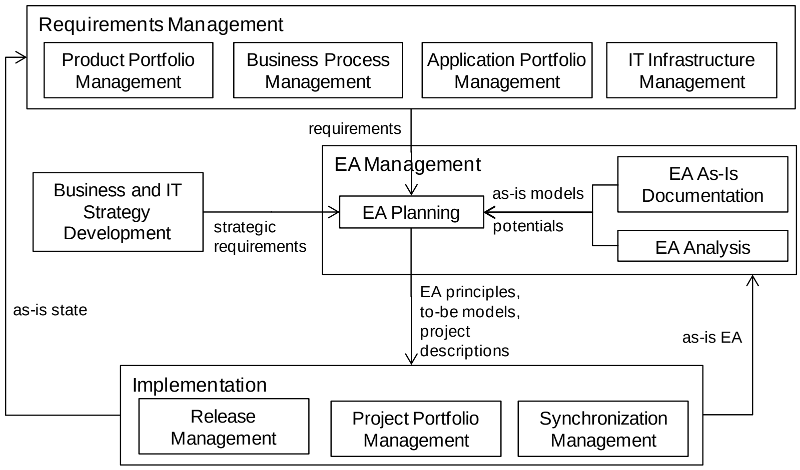


Figure 1: Contextual diagram of the EA planning process [AG10a]

The EA planning process is one of several EAM processes, complementing EA as-is documentation as well as EA analysis processes. EA planning is driven by requirements from the different sub-architectures. As the planning process is based on EA models, information about as-is models and condensed information from EA analysis is needed. Besides that, strategic principles from business and IT strategy processes guide the EA planning process. Main outputs of the planning process are to-be models of future architecture states as well as project descriptions about development projects that will implement the planned enhancements. Those project proposals are accompanied by EA principles that are developed within the EA management processes and distributed into implementation processes via EA planning. The actual implementation is then controlled by release, project portfolio, and synchronization management processes. From there, information about the as-is state of EA is fed back to EA management and to sub-architectures' management processes. In Subsection 3.1 we will review approaches for top-down transformation planning. In Subsection 3.2 we will complement these by approaches for bottom-up planned evolutionary change.

### 3.1 Top-down transformation planning

[AG10a] present a method for EA transformation that puts emphasis on the method's "outputs" as well as on their interdependencies.

The result of the first step of the method is a list of EA elements that need to be changed in order to reach the targeted to-be state (**R3**). Those EA elements are the core subjects of transformation activities. Additionally, as EA elements must not be considered as stand-alone artifacts, information about their relationships to other EA elements needs to be captured as well. Thus, the method produces a list of predecessor/successor relationships linking the as-is with the to-be state, as a basis for plan development activities. These relationships are covered in a transformation model.

The second step plans transformation in detail. The step's result is a description of project proposals (**R4**) containing a list of preconditions, a list of affected EA elements and a relative time line that respects individual life cycles of and temporal interdependencies between EA elements (**R9**). Based on these proposals actual projects can be defined in a third step, taking available budgets and resources into account.

The analysis of differences in the as-is and the to-be state identifies successor relationships between model elements which represent successor relationships of enterprise artifacts, more precisely of their lifecycle phase "in production". From there, the general dependencies of pre-production and decommissioning lifecycle phases can be derived for each segment. The actual sub-division of such pre-production phases varies depending on the type of artifact considered and may comprise various development activities, including specification or testing. Preceding analyses should be grounded in an according information model describing the concepts of the enterprise and the transformation model. [Bu09] have presented such a model introducing the notion of *ProjectAffectable* and applying it to the EA model concepts business application and business support, respectively. Figure 2 shows an adapted version of this model, which also accounts for temporality aspects of EA planning. Thereto, the start and end dates of the projects' work packages are temporal attributes in the sense of the corresponding pattern of [CEF99]. This means that the transformation model can change over time entailing an adaptation of the duration of the work packages (**R10**).

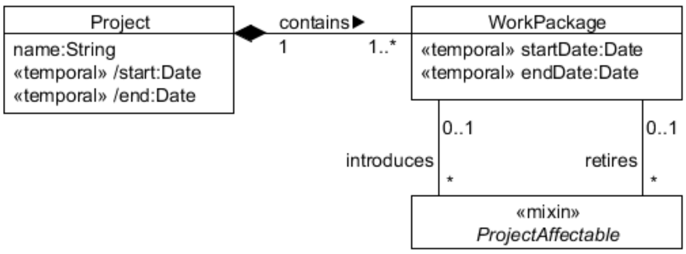


Figure 2: Information model adapted from [Bu09]

The next steps of the EA planning method aim at scheduling an effective sequence of development activities upon the affected artifacts (**R9**). Therefore, the change-induced relationships between the model elements must additionally be taken into account. A temporal restriction, for example, may arise, if a new business process is to be introduced in the to-be state and the supporting applications must be developed before the business process can be activated. Additionally, individual model element lifecycles that might be affected, for example, by vendor support must be respected and integrated.

Scheduling the development activities and establishing the transformation model can be supported by planning techniques from the field of project management. Gantt or milestone charts are commonly used to represent schedules. However, these techniques lack in showing interdependencies between activities and events, which is provided by network techniques [Ke09]. Among such techniques to construct project network diagrams, the Project Management Institute mentions the Precedence Diagramming Method and the Arrow Diagramming Method that are able to display temporal dependencies between individual activities [PMI00]. Based on project network models, formal analyses can be applied to determine actual dates for project plans. A common technique is the Critical Path Method that calculates deterministic early and late start and end dates for each activity, based on an activity-on-node network [AW90].

### 3.2 Bottom-up planned evolutionary change

In classic software development requirements engineering is understood as a process to create and maintain a high-level requirements document for a system (**R1**). According to Sommerville [So01] any requirements engineering process is comprised of four activities. *Feasibility studies* are conducted against the background of an initial set of requirements to answer whether the system under consideration contributes to the overall objective of the organization and whether a system can be implemented within the current technological and economic constraints. Based on the answers to the preceding questions system requirements are subsequently (*requirements elicitation and analysis*) discovered in cooperation with the stakeholders. These requirements are classified and organized into groups of related and mutually coherent requirements. Preceding their documentation in a *requirements specification* document, the requirements are prioritized and negotiated upon. During *requirements validation* different kinds of checks are performed using multiple techniques as systematic or formal review. Thereby, checks for internal validity (*consistency*), external validity (*completeness* and *realism*), and quality (*verifiability*) are performed on the requirements specification.

Proceeding from one-time requirements engineering in a software project to requirements-related processes for a long-living software solution, one comes to the discipline of requirements *management* [So01]. In the single-system environment of classical software engineering the corresponding change management process boils down to three essential activities of *identifying a change proposal, analyzing the proposed change* with respect to costs and feasibility, and *implementing the change*. The IT Infrastructure Library (ITIL) [OGC00] furthers this basic understanding of change management by introducing additional steps for *authorizing change*, which form a counterpart of the feasibility analysis discussed by [So01]. Further, ITIL adds a post implementation review, which in line with classic software engineering may be understood as revising the requirements specification.

Release management is primarily understood as the process of making software available to its users. What at first sight seems to be an easy task is aggravated by the complex interdependencies of modern component-based software. [HW03] discuss five requirements that release management has to satisfy: *1)* make component dependencies explicit, *2)* keep releases consistent with respect to these dependencies, *3)* control user access to a release, *4)* support development team transparently, and *5)* store history of releases and their usages. Key concept of release management is component metadata, i.e. specifications on dependencies between components, but also on dates of release and of withdrawal.

After an internally and externally validated set of requirements has been converted to an agreed project portfolio (**R4**), synchronization management aims at the interdependencies between the changed projects (**R9**). [ES10] describes a method for identifying such dependencies exemplified along the following three types of dependencies: *1) organizational dependencies*, resulting from multiple projects targeting the same part of the organization at the same time, *2) interface dependencies*, resulting from the need to migrate applications' interfaces simultaneously with changing the related applications, and *3) technical dependencies*, resulting from infrastructure dependencies of applications, e.g. the compatibility with only distinct operating systems.

## 4 An integrated method for EA planning

The artifact of the paper at hand is an integrated method [MS95] for EA planning taking global top-down driven transformation plans as well as local bottom-up driven evolution of partial architectures into account.

A method consists of a sequence of design activities which are related to a procedure model. The produced design results of the design activities are represented by an information model. Additionally, a method consists of roles which describe the participants involved in the design activities. At the same time the inclusion of roles introduces various perspectives different stakeholders may have on design activities. Instructions on how the design results are produced and documented are provided by adequate techniques [Br05], [Gu94].

Our analysis of related work shows that all of the relevant requirements (Section 2) but (**R6**) are addressed by preliminary or related work described in Section 3. Our analysis also shows that general planning cycles for global EA transformation plans on the one hand side and more local requirements, release, and synchronization management on the other hand side are similar. While the granularity of their planning objects differs (and thus the approaches' formality) their activities and techniques are similar. The main challenges for an integrated planning approach are its organization and coordination as well as its consistent description in an information model (**R6**). Therefore, this contribution focuses on the coordination of design activities (Subsection 4.1) and on an integrated information model (Subsection 4.2).

### 4.1 Coordination mechanisms

Deriving consistent and integrated plans for EA transformation and evolution is a complex task because

- a number of subsystems (represented e.g. by architectural layers) need to be coordinated and the subsystems are again complex,
- a number of stakeholders responsible for one or more subsystems need to be coordinated, thus the task can only be solved by division of labor,
- neither the (sub) system(s) nor its environment(s) can be expected to remain unchanged during a planning cycle and a plan's implementation.

It is unlikely to develop a consistent and integrated plan for EA transformation and evolution in a single iteration by central planning. Instead several iterations of centralized and distributed planning will be necessary.

In cybernetics this situation is modeled as a closed-loop control system comprised of goal setting, (meta-)planning, controlling, running, and measuring. The measurement results are fed back to the controller who then adjusts the system's input. If engineered properly closed-loop control systems achieve defined goals even in a dynamic environment. In case a system consists of several other systems it can be modeled as a system of intermeshed closed-loop control systems (Figure 3) [HG94]. In such a situation running system A is equivalent to (meta-)planning and controlling of system B.

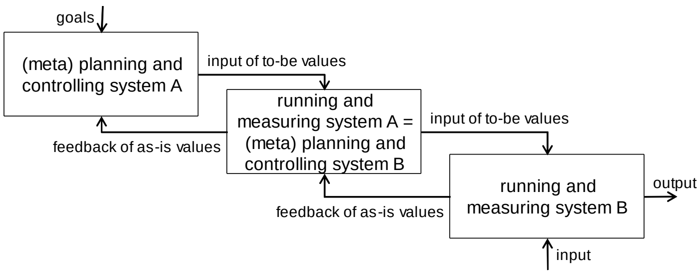


Figure 3: Closed-loop control systems based on [HG94]

Systems of intermeshed closed-loop control systems are guided and controlled by *1)* overall goals and *2)* contracts describing the interfaces between each closed-loop control system. Overall goals are defined by business strategy, IT strategy, and EA strategy in particular. The structure of interfaces is defined by the governance structure relevant for the participants of the planning cycle. Legitimacy of each partial plan as well as the integrated solution can be evaluated based on existing EA principles. Such systems can be vertically or horizontally intermeshed. Vertically intermeshed systems can be found in hierarchical relations, e.g. overall application planning is vertically intermeshed with application planning for divisions, business units, regions, products etc. Horizontally intermeshed systems can be found with peer-level planning, e.g. application planning is horizontally intermeshed with IT infrastructure planning.

As the examples above illustrate, the general approach to design an integrated EA planning function as a system of intermeshed closed-loop control systems allows for a variety of configurations. According to the Conant-Ashby theorem "every good regulator of a system must be a model of that system" [CA70] which means that the effectiveness of the individual configuration depends on how well it represents the actual interdependencies of the planning objects. The following information model ensures that the theorem is fulfilled. In Section 5 we will discuss an example configuration.

### 4.2 Information model

Figure 4 shows an information model that is capable of describing the transformation model for EA planning in an embracing manner.

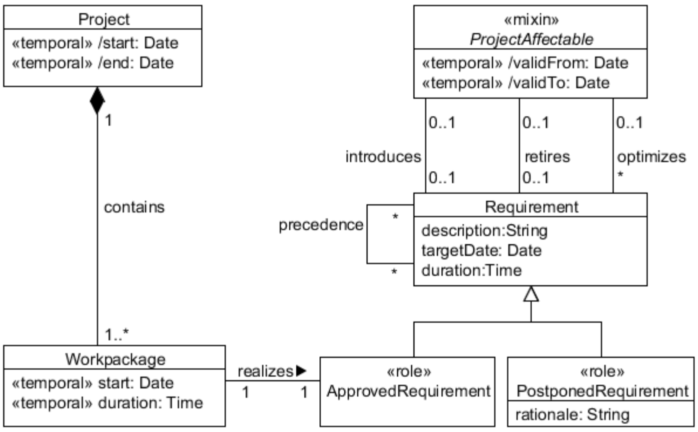


Figure 4: Information model for describing the EA planning transformation model

Central concept of the information model is the *mixin* PROJECTAFFECTABLE which marks a concept from an EA information model as subject to EA planning, thereby allowing a using organization to decide which concepts, e.g. business applications, are considered in EA planning.

Beyond the basic linkage between the concept REQUIREMENT and the concept WORKPACKAGE in Figure 2, the comprehensive information model introduces a different conception. Firstly, a requirement reflects an *intended* change in contrast to a work package that describes a *scheduled* change. Secondly, complementing the relationships *introduces* and *retires* that target changes on an EA-relevant level, the requirement provides the relationship *optimizes*. This relationship allows describing maintenance requirements that do not reshape the enterprise on an EA-relevant level. Together, the three types of relationships and the relationship *precedence* cover all types of dependencies between architecture elements in a uniform manner (see requirements (**R3**) and (**R6**)). In particular, they supply a basis for a technique for comparing an as-is and a to-be state (**T1**). Further, each requirement bears a target date, i.e. a date as of which the change is to be effective, as well as an expected duration for implementation. Based on this information, the technique of a critical path analysis (**T5**) can be implemented. Above information not only supplies concepts for documenting requirements, but also for documenting decisions on requirements. A requirement can on the one hand switch to the role APPROVEDREQUIREMENT, where it is further linked to a work package for implementation. On the other hand, a requirement may also be converted to a POSTPONEDREQUIREMENT to reflect that currently no plan for resolving the identified conflicts can be scheduled. Thereby, techniques for requirements management (**T2**) as well as techniques for synchronization management (**T3**) are supported via the temporal and explicit dependencies between the requirements. Table 1 summarizes design results, information model, design activities, and techniques of our method for an integrated EA planning.

Table 1: Method summary

| | **Method for planning enterprise transformation** |
|---|---|
| **Design Results** | *DR1*. Meta result [or input]: System of local and global partial plans (**R1**)<br>*DR2*. Meta result [or input]: Coordination mechanisms (**R6**)<br>*DR3*. Meta result [or input]: Principles and goal structures (**R6**)<br>*DR4*. List of model elements that need to be changed in each partial plan (ProjectAffectables) (**R3**)<br>*DR5*. Information about relationships to other model elements and other partial plans (requirement relations) (F)<br>*DR6*. Description of project proposals containing work packages realizing approved requirements (**R4**, **R9**) |
| **Information Model** | *I1*. Transformation Model (**R3**, **R6**; provides basis for **T1-3**, **T5**) |
| **Design Activities** | *A1*. Analyze differences between as-is and to-be (requirements) for each local and global planning iteration (DR4)<br>*A2*. Synchronize partial plans and identify inconsistencies (DR5)<br>*A3*. Separate segments, i.e. identify work packages and projects (DR6)<br>For each segment:<br>*A4*. Identify general temporal interdependencies (DR6)<br>*A5*. Schedule an effective sequence of development activities, by refining temporal interdependencies (DR6)<br>*A6*. Draw network plan displaying all interdependencies (DR6) |

| | |
|---|---|
| **Techniques** | *T1*. Graph comparison (for A1)<br>*T2*. Techniques of requirements management (for A1)<br>*T3*. Techniques of synchronization management (for A2, A3)<br>*T4*. Establishing project network model/graph using Precedence Diagramming Method and the Arrow Diagramming Method (for A5)<br>*T5*. Deriving a project timeline using Critical Path Method (for A5) |

## 5 Demonstration

In this section, we demonstrate (and initially evaluate) our approach employing an industry case illustrating one possible configuration of a system of intermeshed closed-loop control systems as proposed in this paper. The using company provides banking solutions, offering a banking platform to private and universal banks. The company's EAM focuses on three main fields: Application development, application management, and operations. The application development division is responsible for the development of the banking platform which is described by an application architecture model, an IT architecture model, and an operating model.

Challenges within the architectural development plan are the coordination of activities of development teams (**R6**), assurance that all dependencies are addressed, and that milestones of integration and development activities are met simultaneously (**R9**). If, for example, a component of an application needs an interface to a component of another application (**R3**) at a certain time for a certain milestone (e.g. test or release), it has to be assured that both components are available at that very point in time. This simple example grows very complex as the banking platform comprises over 200 applications, each consisting of a multitude of components that each has its own lifecycle.

The planning activities at the company aim at the further development of their core product, a banking platform, which consists of various applications, interfaces and middleware components. As the company also provides the hosting of the platform, infrastructure is an issue, too. For this perspective, the company's information model covers the relationships between business applications, their constituting components and the underlying hardware devices (see Figure 5). The three conceptual classes representing architecture elements are thereby considered *project affectable*, i.e. are subject to the dependence planning mechanisms described in our method.


Figure 5: Information model of the demonstration case

In such a scenario both, top-down transformation plans and the bottom-up evolution of partial architectures occur and need to be integrated (**R6**). Further development of the banking platform (high-level transformation plan) is driven by a superordinate planning team that defines the general development strategy of the platform and the relevant principles (**DR1**, **DR2**, **DR3**). At the same time requirements for the development of the individual components and sub-architectures are collected separately. The configuration of the underlying intermeshed closed-loop control system is a matrix structure defined by lead architects (business, applications, IT, operations, and security) and domain architects responsible for business domains structured by business products (**DR1**, **DR2**).

Partial architecture roadmaps are defined following the general EA guidelines. The roadmaps consist of models representing snapshots of the desired architecture for up to three future points in time, taking into account existing vendor specific constraints if applicable (**DR4**). Individual component development plans are then integrated into these roadmaps. The coordination between high-level transformation plans and local roadmaps of evolution is ensured by the matrix structure. While domain architects control the development and realization of high-level transformation plans, lead architects drive and ensure the development of components taking detailed restrictions into account. In joint workshops all parties consolidate their plans based on EA goals and principles. Possible alternatives are thereby discussed and evaluated (**DR5**). This particular matrix configuration uses vertical and horizontal coordination of the meshed systems. Vertical coordination occurs e.g. by aligning a domain specific transformation plan with the detailed plans for applications of that domain. Horizontal coordination occurs e.g. by aligning the partial architectures of the lead architects.

To determine a sequence of development activities general milestones and components to be developed are defined. Afterwards, the development phases of the elements' lifecycles are planned in detail, i.e. in terms of specification and testing phases. Upon those specifications a rough project program schedule can be defined (**DR6**). Finally, on the basis of the project outlines actual project planning is enabled. For that purpose, the project proposals can be enriched with information about costs, quality metrics, staff, risks, and resources.

The case description illustrates that an appropriate configuration of a system of intermeshed closed-loop control systems that takes the concerns of relevant stakeholders (domain and lead architects) into account is a efficient solution for coordination of top-down transformation planning and bottom-up evolutionary change. Whether the results of such a planning approach justify the respective efforts cannot be shown here. Therefore, the case study does not serve as a full evaluation. However, discussions with the case study company showed that – apart from a well defined plan – such an approach has a number of positive side effects – most notably stakeholders have a common high level of knowledge about the organization and are therefore able to effectively coordinate each other before or during actual projects.

## 6 Reflection and outlook

In this paper, we analyzed top-down driven methods for EA transformation planning as well as bottom-up driven methods for requirements, release, and synchronization management. We integrated design activities and techniques used in these domains into a method for an integrated EA planning. Our method description focuses on the coordination of distributed planning activities and an underlying information model, which provides the modeling concepts for addressing requirements (**R1**), (**R3**), and (**R6**).

Building on the contained information about the transformation and evolution plans, the delineated method and techniques particularly aim at requirements (**R4**) and (**R9**). The applicability of method and information models has finally been demonstrated with an industry case.

However, there still are a number of questions that we have not answered yet but given an indication in the case only. For example our approach does not generally describe how exactly configuration mechanisms for systems of intermeshed closed-loop control systems work and whether there may be a (limited) number of reference configurations for an integrated EA planning

Also criteria for determining the number of iterations for global and local planning cycles are unknown. It is further unknown, which exit conditions are valid, i.e. when to stop the EA planning phase and proceed with refinements and implementations.

Finally, it has to be shown that there is a business case for such a "coordinated" approach of an integrated EA planning opposed to an even more decentralized and less coordinated development plan. However, in practice we observe a growing demand and acceptance for such a coordinated planning. This is because alternatively a coordination of projects applying for funding is often performed by prioritization and projects are postponed after available budgets are assigned. The prioritization process, taking only the financial dimension into account, however, often results in dysfunctional plans.

## Bibliography


[AG10a] Aier, S.; Gleichauf, B.: Applying design research artifacts for building design research artifacts: A process model for enterprise architecture planning. In (Winter, R.; Zhao, J.L.; Aier, S.): (eds.) Global Perspectives on Design Science Research, 5th International Conference, DESRIST 2010, Springer, 2010; pp. 333-348.

[AG10b] Aier, S.; Gleichauf, B: Towards a systematic approach for capturing dynamic transformation in enterprise models. In HICSS, 2010; pp. 1-10.

[AW90] Antill, J.; Woodhead, R.: Critical path methods in construction practice. Wiley-Interscience, 1990.

[Br05] Braun, C.; Wortmann, F.; Hafner, M.; Winter, R.: Method construction - a core approach to organizational engineering. In SAC, 2005; pp. 1295-1299.

[Bu09] Buckl, S.; Ernst, A.M.; Matthes, F.; Schweda, C.M.: An information model for managed application landscape evolution. Journal of Enterprise Architecture (JEA) 5(1), 2009; pp. 12-26.

[CA70] Conant, R.C.; Ashby, W.R.: Every good regulator of a system must be a model of that system. Intl. J. Systems Science, 1970; pp. 89-97.

[CEF99] Carlson, A.; Estepp, S.; Fowler, M.: Temporal patterns. In Pattern Languages of Program Design. Addison Wesley, Boston, MA, USA, 1999.

[ES10] Ernst, A.M.; Schneider, A.W.: Roadmaps for enterprise architecture evolution. In Engels, G.; Luckey, M.; Pretschner, A.; Reussner, R.: (eds.) 2nd EuropeanWorkshop on Patterns for Enterprise Architecture Management (PEAM2010) Paderborn, Germany, 2010; pp. 253-266.

[Gu94] Gutzwiller, T.A.: Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. Ph.D. thesis, Universität St.Gallen, 1994.

[HG94] Hahn, D.; Grünewald, H.: PuK, Controllingkonzepte: Planung und Kontrolle, Planungs- und Kontrollsysteme, Planungs-und Kontrollrechnung. Gabler, 1994.

[HV93] Henderson, J.C.; Venkatraman, N.: Strategic alignment: Leveraging information technology for transforming organizations. IBM Systems Journal 32(1), 1993; pp. 472-484.

[HW03] van der Hoek, A.; Wolf, A.L.: Software release management for component-based software. SOFTWARE-PRACTICE AND EXPERIENCE 33(1), 2003; pp. 77-98.

[IOS07] International Organization for Standardization.: ISO/IEC 42010:2007 Systems and software engineering - Recommended practice for architectural description of software-intensive systems, 2007.

[Ke09] Kerzner, H.: Project management: a systems approach to planning, scheduling, and controlling. Wiley, 2009.

[LLO93] Luftman, J.N.; Lewis, P.R.; Oldach, S.H.: Transforming the enterprise: The alignment of business and information technology strategies. IBM Systems Journal 32(1), 1993; pp. 198-221.

[MS95] March, S.; Smith, G.: Design and natural science research on information technology. Decision Support Systems 15(4), 1995; pp. 251-266.

[Ma08] Matthes, F.; Buckl, S.; Leitel, J.; Schweda, C.M.: Enterprise Architecture Management Tool Survey 2008. Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2008.

[OGC00] Office of Government Commerce (OGC).: ITIL - Service Delivery. IT Infrastructure Library (ITIL), The Stationery Office, Norwich, UK, 2000.

[PMI00] Project Management Institute.: A Guide to the Project Management Body of Knowledge, 2000.

[Pe07] Peffers, K.; Tuunanen, T.; Rothenberger, M.; Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 2007; pp. 45-77.

[Pu06] Pulkkinen, M.: Systemic management of architectural decisions in enterprise architecture planning. four dimensions and three abstraction levels. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006; pp. 179c.

[RWR06] Ross, J.W.; Weill, P.; Robertson, D.C.: Enterprise Architecture as Strategy. Harvard Business School Press, Boston, MA, USA, 2006.

[So01] Sommerville, I.: Software Engineering. Pearson Studium, Munich, Germany, 6th edn., 2001.

[Wa05] Wagter, R.; van den Berg, M.; Luijpers, J.; van Steenbergen, M.: Dynamic Enterprise Architecture: How to Make IT Work. John Wiley, 2005.

[Wi08] Winter, R.: Business engineering - betriebswirtschaftliche konstruktionslehre und ihre anwendung in der informationslogistik. In: Integrierte Informationslogistik. Springer, Berlin, Heidelberg, Germany, 2008; pp. 17-38.