

Goal Based Alignment of Enterprise Architectures

Balbir S. Barn and Tony Clark

School of Engineering and Information Sciences, Middlesex University, London NW4 4BT, U.K.

Keywords: Enterprise Architecture, Simulation, Modeling.

Abstract: Business and IT alignment remains an ongoing concern for organizations. In this paper, we propose a set of technologies and concepts - notably goals and computable functions which can be used to provide a measure of equivalence between as-is and to-be enterprise architectures.

1 INTRODUCTION

Business and IT alignment has remained an ongoing concern for organisations since the 1980s (Luftman, 2004). Throughout this period, researchers have addressed the importance of alignment and in particular the need for congruence between business strategy and IT strategy (Chan and Reich, 2007). The *Strategic Alignment Model* (SAM) (Henderson and Venkatraman, 1993) defines that alignment is the degree of fit and integration among business strategy, IT strategy, business infrastructure, and IT infrastructure.

Enterprise Architecture (EA) aims to capture the *what, why* and *how* of a business including the alignment between business functions and IT systems. Despite the opportunities presented for addressing Business and IT Alignment (BIA) using EA, the current state of the art of EA presents issues such as: large unwieldy methods such as those derived from TOGAF (Spencer et al., 2004); a lack of precision in the methods. This paper contributes a rigorous approach to aligning business goals to IT infrastructure using functions that can be applied to both as-is and to-be enterprise architecture models.

2 RELATED WORK

A model for measuring BIA was originally proposed by Henderson and Venkatraman, the Strategic Alignment Model (SAM) (Henderson and Venkatraman, 1993). Other researchers have extended the SAM, e.g. Luftman proposed the Strategic Alignment Maturity Model (SAMM) measures maturity levels of strategic alignment along several key dimensions such as governance and skills. and categorizes levels

into strategic, tactical and operational states (Luftman, 2004). Vargas et al. have produced a model that has been consolidated from a review of existing models and have studied the use of the model in case studies of public universities in Nicaragua (Vargas Chevez, 2010). Other models for strategic alignment have focused on executive feedback measures (Avison et al., 2004), organizational structure (Bergeron et al., 2004), social dimensions such as vision (Reich and Benbasat, 1996) and cognitive dimensions (Tan and Gallupe, 2006). Chung et al. recommend that new research should examine the recursive relationship between alignment and the extent of applications implementation and IT infrastructure flexibility (Chung et al., 2003). Street has begun this work, examining 'service gaps' as a measure of alignment (Street and Denford, 2012). One of the emerging standardisation efforts on goal modelling is the Business Motivation Model (BMM)¹. This model provides a set of concepts for describing business plans including links to other notations such as BPMN.

3 LEAP

The LEAP modelling approach is an integrated model, method and simulation environment that uses concepts from component based design, event driven architecture and service oriented architecture. The approach has been described in detail elsewhere (Clark et al., 2011; Clark and Barn, 2011; Clark and Barn, 2012).

In order for an organization to understand the impact of a business goal both in terms of current tech-

¹<http://www.omg.org/spec/BMM/1.1/PDF/>

nology and likely changes, the organization must first describe the business goal using the LEAP DSL. The goal is described using three basic elements: (1) a description of the goal and quality attribute that captures the essence of the goal; (2) a set of assumptions that are either invariant or are values that are indicative of the goal being met; (3) a function F from system execution traces to a partially ordered domain that can be used as a measure of goal satisfaction.

Our proposal is that architectural alignment with respect to a business goal must be expressed in terms that can be precisely measured. Our claim is that any meaningful business goal can be recast as a function over the architectures; if the values produced by the function are ordered such that the ordering corresponds to goal satisfaction then we can use this to test whether the to-be architecture is in some way *better than* the as-is architecture with respect to the business goal.

4 CASE STUDY AND EVALUATION

Higher education institutions (HEI) in the UK are faced with a challenging and dynamic business environment where public funding of HEIs has been reduced by up to 70%. This lost funding is being replaced by the introduction of a new student fees regime beginning in 2012 following a bill introduced in the UK parliament in November 2010. This environment places greater expectations on operational efficiency and one mechanism for addressing this is to ensure that business strategy and IS/IT strategy is better aligned (Gregor et al., 2007).

The University of Scrabbleshire (UoS), has decided that they will introduce student fees at the maximum permitted level of £9000. However, they recognise that such high fees will have to be justified so the Corporate Plan has introduced new requirements. The University will compete on quality in terms of research outputs. Such a change will hopefully lead to a higher ranking in the research league tables.

To address these business goals, UoS recognises that it needs to go through a number of stages, firstly it will have to assess the current systems and processes and how they support the business goals. Some measure of “quality” or “extent” of how these goals have been met will have to be derived. Currently, there is no mechanism for approaching this measurement capability. A second stage is to identify changes to the systems, perhaps identify new systems and or business processes. A third stage is of course to measure how the proposed changes have addressed these

business goals.

During the assessment of the existing systems, the IT services department working under the leadership of Dr Magnus Sungam, identify the following systems that are potentially impacted in some form by the new corporate plan: (1) timetabling systems that schedule staff to specific modules to specific room (2) the workload management system used by Heads of Department to ensure that staff are allocated work in an appropriate manner; (3) the University Repository that provides external access to the research outputs of the University; (4) the HR system that manages contracts for staff; (5) the University Website.

The various systems identified need to change in a number of ways in order to support the business goals laid out in the corporate plan. Various changes are envisaged. The workload management system will need to differentiate and support workload planning for different categories of staff (those that are research active and those that are teaching only). The timetabling system will need to integrate new constraints that ensure that research active staff have at least and preferably three clear days to support research activity. Both these systems will need to extract information from the HR contracts system. The University Repository will need to be integrated with the external facing Corporate website so that changes to an academic’s research outputs is updated automatically on the website to reflect updates to the Repository. A new system to manage the research process for preparing to the UK national research assessment process for ranking UK universities will need to be developed.

5 ASSUMPTIONS

There are a collection of business assumptions and directives that must be defined before the goal can be represented and analysed. A directive is something that must hold at all times, for example: (1) In planning any changes to UoS no new staff resources can be made available; (2) All changes must maintain the current levels of teaching.

Each assumption allows us to implement a change without taking the wider context into account when analyzing or measuring the effect of the change in terms of the overall goals, for example: (1) Reputation will be enhanced by visibility of research outputs via the UoS web-site; (2) UoS is effective at measuring the quality of its research outputs; (3) Research is to be measured on a 4-point scale with 1 being the lowest level of quality where the likelihood of producing high-quality research is increased by providing staff with a contiguous period of time free from

teaching and admin responsibilities; (4) The number of research outputs and their quality are positively correlated to reputation.

6 SPECIFICATION: AS IS

The LEAP component specifications provide operations that allow the UoS to be simulated in terms of its research activities. This section provides an overview of the UoS LEAP specification as a single component `scabbleshire` that contains sub-components and an operation `perform` that is used to define the simulation:

```
component scabbleshire {
  component personnel { model { class Staff { name:str } } }
```

The first component, shown above, is `personnel` that maintains a database of all staff employed by the university. Currently UoS employs all academic staff on the same type of contract. The next component is used to record research activity:

```
component research_activity {
  model { class Activity { staff:str; quality:int; journal:str } }
  port actions[in]: interface {
    start_research(staff:str, journal:str):void;
    perform_research(staff:str):void;
    abort_research(staff:str):void }
  port produce[out]: interface {
    research(staff:str, quality:int, journal:str):void }
  spec {
    abort_research(name:str):void {
      pre Staff(name,quality,journal) ?(quality>0)
      post Staff(name,0)
      messages produce <- research(name,quality,journal) }
    start_research(name:str, journal:str):void {
      pre not(Staff(name,_,_))
      post Staff(name,0,journal) }
    perform_research(name:str):void {
      pre Staff(name,a1,journal) ?(a1 < 3)
      post Staff(name,a2,journal) ?(a2 = a1+1) }
    perform_research(name:str):void {
      pre not(Staff(name,a1,journal))
      post Staff(name,0,journal) }
    perform_research(name:str):void {
      pre Staff(name,3,journal)
      post not(Staff(name,_,_))
      messages produce <- research(name,4,journal) } } }
```

The `research_activity` component manages a database that maintains information about staff research activity. An `Activity` record includes the quality of the research (on a scale of 0 to 4) and the `Journal` to which it will be submitted if completed. The input interface `actions` support operations for starting a research activity, performing research and aborting the research. After initiating a piece of work, a member of staff can perform up to 4 increments at which point the research is ready for submission. At any time the research can be aborted, at which time it is assumed to be submitted as-is providing *some* work has been undertaken.

A head of department manages a resource database that allocates each member of staff to teaching, research and administration tasks:

```
component resource_planning {
  model {
    class Staff { name:str; teaching:int; research:int;
                 admin:int } }
  invariants {
    time_100 {
      forall Staff(_,teaching,research,admin)
      in state { teaching + research + admin = 100 } } }
```

The UoS uses a centralized room booking system. The simulation uses the room booking system to determine whether a member of staff can undertake research:

```
component room_booking {
  model { class Booking { staff:str; room:int; time:int } } }
```

The UoS research repository contains a record of research outputs produced by members of staff. We will assume that all submissions added to the repository are in print and that the research quality depends on the length of time spent on the research. The operation `research` is used to update the repository:

```
component repository {
  model {
    class Entry { staff:str; entry:int; quality:int; journal:str } }
  port deposit[in]: interface {
    research(name:str, quality:int, journal:str):void }
  spec {
    research(name:str, quality:int, journal:str):void {
      post
      Entry(name,entry,quality,journal)
      not(Entry(_,entry,_) in state@pre) } } }
```

The UoS web site is the public facing interface for UoS. It contains many types of entry but for the purposes of the simulation it is important to know whether a repository entry is referenced on the web site:

```
component web { model { class Entry { repository_id:int } } }
```

The `scabbleshire` component is completed by defining the data and operations used for the simulation. The simulation is driven by a collection of time ordered messages that are delivered to the sub-component defined above. Each message has a time, name and some argument data:

```
model {
  class Message { time:int; name:str; args:[Value] }
  class Time { time:int } }
```

The simulation is executed by the `perform` operation that delivers messages. A message is ready for delivery when its time becomes current, if there are no messages ready then time advances:

```
spec {
  perform():void {
    pre Time(t) not(Message(t,message,args))
    post Time(tt) ?(tt = t + 1) } }
  ... further specifications below...
```

There are three different messages: `start` that attempts to initiate research by a member of staff; `do_research` that attempts to make an incremental research step; `transfer` that ensures that all repository entries for a member of staff are transferred to the

UoS web site. When research starts the appropriate message is sent to the `research_activity` module:

```
perform():void {
  pre Time(t) Message(t,'start',[name,journal])
  post not(Message(t,'start',[name,journal]))
  messages research_activity.actions
  <- start_research(name,journal) }
```

When the simulation attempts to direct a member of staff to do research it may be prevented from succeeding because they are timetabled for teaching, otherwise a message is sent to the `research_activity` component:

```
perform():void {
  pre Time(t) Message(t,'do_research',[name])
  ?(!exists Staff(name,_,t) in room_booking.state)
  post not(Message(t,message,args))
  messages research_activity.actions <- perform_research(name)
}
perform():void {
  pre Time(t) Message(t,'do_research',[name])
  ?(exists Staff(name,_,t) in room_booking.state)
  post not(Message(t,message,args)) }
```

When the simulation attempts to direct a member of staff to transfer their research from the repository to the UoS web site, teaching can prevent the transfer:

```
perform():void {
  pre Time(t) Message(t,'transfer',[name])
  ?(!exists Staff(name,_,t) in room_booking.state)
  post ?(forall Entry(name,id,_) in repository.state {
    exists Entry(id) in web.state })
}
perform():void {
  pre Time(t) Message(t,'transfer',[name])
  ?(exists Staff(name,_,t) in room_booking.state)
  post not(Message(t,message,args)) }
```

Our assumption is that effective research needs a contiguous amount of free time slots. Therefore, when staff are scheduled for teaching, current research is aborted (i.e. submitted as-is):

```
perform():void {
  pre Time(t) Staff(name,_,t) in room_booking.state
  messages room_booking.actions <- abort_research(name) }
```

The global invariants ensure that all staff managed by the resource planning and room booking components are registered with personnel:

```
invariants {
  resourced_staff {
    forall Staff(name,_,_,_)
    in resource_planning.state {
      exists Staff(name) in personnel.state } }
  booked_staff {
    forall Staff(name,_,_) in room_booking.state {
      exists Staff(name) in personnel.state } }
```

Finally, the components are connected in the initialization clause of `scrabbleshire`:

```
init {connect(research_activity.produce,repository.deposit)}
```

7 SEMANTICS

The operational semantics of a LEAP specification is defined in figure 1. A component κ is represented as

(c, Δ, S) where c is the component identifier, Δ is the component database and S is a set of operation specifications. LEAP allows components to be nested; the specification flattens this tree structure. LEAP supports invariants on each component that are assumed to hold at all times within a component κ . Finally, LEAP ports are not represented in the semantics and port connections are handled by directing output messages to the target component.

An operation specification $m(\tilde{i})_{p,q}^\mu \in S$ consists of an operation name m , a sequence of formal parameter names \tilde{i} , a precondition p , postcondition q and a predicate μ . The specification requires that when the operation named m is invoked in response to processing a message with the same name, if p is true of the pre-state of the component then q is true of the post-state and μ defines the messages that are sent.

The semantics in figure 1 defines a relationship $M \vdash C \Rightarrow C', M'$ that defines an execution of components C with input messages M producing a new collection of components C' with a mixture of unprocessed messages and new output messages M' . Rule L-1 is the workhorse that defines how a single step is performed by processing a message. The precondition must hold for the pre-state Δ in the context of the supplied argument values \tilde{v} and values for the free variables $FV(p)$, the postcondition must hold for both the pre-state (referenced as `state@pre`) and the post-state Δ' with respect to the argument values, the same values used for free variables in the precondition, and values for the free variables in q . Finally the message predicate μ must hold for the output messages in the context of all the previous variable bindings.

Rule L-2 just lifts the single step defined in L-1 to sequences of messages and sets of components. Note that the new output messages in L-2 (M_1) are added to the end of any unprocessed messages (M_0). Rule L-2 defines how sequences of execution are concatenated.

We define a relationship $C_0 \rightarrow_M T$ that holds between a collection of UoS components C_0 , a set of message terms $M = \{\text{Message}(t, n, a), \dots\}$ and an execution trace $T = [C_0, C_1, \dots, C_n]$ such that $[\text{perform}()] \vdash C_0 \Rightarrow C_n, \emptyset$ and all messages M have

$$\text{L-1} \frac{p[\tilde{v}_1/\tilde{i}][\tilde{v}_2/FV(p)](\Delta) \quad q[\tilde{v}_1/\tilde{i}][\tilde{v}_2/FV(p)][\tilde{v}_3/FV(q)](\Delta, \Delta') \quad \mu[\tilde{v}_1/\tilde{i}][\tilde{v}_2/FV(p)][\tilde{v}_3/FV(q)][\tilde{v}_4/FV(\mu)](\Delta, \Delta', M)}{m_c(\tilde{v}_1) \vdash (c, \Delta, m_{(p,q)}^\mu(\tilde{i}) : S) \Rightarrow (c, \Delta', m_{(p,q)}^\mu(\tilde{i}) : S), M}$$

$$\text{L-2} \frac{m \vdash \kappa \Rightarrow \kappa', M_1}{m : M_0 \vdash \{\kappa\} \Rightarrow \{\kappa'\}, M_0 + M_1}$$

$$\text{L-3} \frac{M_1 \vdash C_1 \Rightarrow C_2, M_2 \quad M_2 \vdash C_2 \Rightarrow C_3, M_3}{M_1 \vdash C_1 \Rightarrow C_3, M_3}$$

Figure 1: LEAP Semantics.

$$\begin{aligned}
 F_1(- + [- \cup \{(R, \Delta)\}]) &= \#\Delta \\
 F_2(- + [- \cup \{(R, \Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4)\}]) &= \sum_{i=1,4} \#\Delta_i * i \\
 F_{3a}([- \cup \{(P, \Delta_b), (R, \Delta_r)\}]) &= \#\{n \mid \text{Staff}(n) \in \Delta_b, \text{Entry}(n, -, -) \in \Delta_r\} \\
 F_{3b}^s(\tau_1 : \tau_2 : \tau_3 : \tau_4 : T) &= \begin{cases} 1 + F_{3b}^s(T) & \text{when } \bigwedge_{i \in 1,4} \neg \text{teaching}(s, \tau_i) \\ F_{3b}^s(\tau_2 : \tau_3 : \tau_4 : T) & \text{otherwise} \end{cases} \\
 F_{3b}^s(\emptyset) &= 0 \\
 F_{3b}(T) &= \sum_{s \in S} F_{3b}^s(T) \\
 F_3(T) &= F_{3a}(T) + F_{3b}(T) \\
 F_4(T + [- \cup \{(web, \Delta)\}]) &= (\#\Delta) / (\#T + 1) \\
 F(T) &= F_1(T) + F_2(T) + F_3(T) + F_4(T)
 \end{aligned}$$

Figure 2: Goal function.

been processed.

8 GOAL FUNCTION

The UoS business goal can be implemented as a measure F on execution traces: (F_1) the number of research outputs in the UoS repository, our assumption is that it is necessary to increase the number of outputs in order to increase reputation; (F_2) the weighted total number of research outputs, reputation is increased by maximizing the locally attributed quality of each output; (F_3) maximizing staff engaged in research decomposing into F_{3a} and F_{3b} ; (F_{3a}) the total number of academic staff engaged in research; (F_{3b}) the number of times that each member of academic staff has an opportunity to produce an output of the maximum level of quality; (F_4) minimizing the time spent to produce a given number of outputs.

The measure F is defined in figure 2 where the following assumptions are made: R=repository and P=personnel; # produces the size of a set or sequence; sequences (of messages) are concatenated using +; the set Δ_i is the set of terms of the form $\text{Entry}(n, e, i, j)$; the set S used in the definition of F_{3b} is the set of all staff names. Given a simulation defined by a collection of messages M and a UoS specification C then $C \rightarrow_M T$. The measure of how well C meets the business goal is represented by $F(T)$.

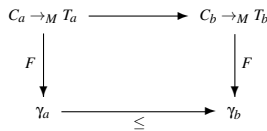


Figure 3: Goal alignment.

9 CONFORMANCE

Figure 3 shows how this machinery is intended to support the comparison of business architectures, by producing a measure γ_a for a specification C_a (the *as-is*

architecture) and a measure γ_2 for the specification C_b (the *to-be* architecture), for the same simulation messages M . Since each measure is an integer, they can be compared with \leq and we claim that C_b is better at satisfying the business goal than C_a is the diagram commutes.

10 SPECIFICATION: TO BE

The model for personnel is updated and staff are moved on to different types of contracts. Staff are either research active, in which case they are expected to spend at least 50% of their time on research activities, or are teaching only in which case they are allocated no time for research:

```

model {
  class Staff { name:str }
  class Contract { id:int }
  class TeachingOnly extends Contract { }
  class ResearchActive extends Contract { }
  assoc Legal { staff Staff contract Contract }
}
    
```

The repository is updated to generate an event each time an update is made. Any UoS component can monitor the event stream in order to detect the update:

```

component repository {
  model {
    class Entry { staff:str; entry:int; quality:int; journal:str }
  }
  port deposit[in]: interface {
    research(name:str, quality:int, journal:str):void
  }
  port events[out]: interface {
    update(entry:int):void
  }
  spec {
    research(name:str, quality:int, journal:str):void {
      post Entry(name, entry, quality, journal)
      not(Entry(_, entry, _) in state@pre)
      messages events <- update(entry) } }
}
    
```

The web site is updated to monitor repository events:

```

component web {
  model { class Entry { repository_id:int } }
  port monitor[in]: interface { update(entry:int):void }
  spec { update(entry:int):void { post Entry(entry) } }
}
    
```

A new component is introduced that manages a database of quality measures for journals. This will be used to override local quality measures for research outputs:

```

component quality {
  model { class Journal { name:str; quality:int } }
}
    
```

A global invariant is added that enforces the UoS directive that teaching only staff will not be given time for research activities:

```

teaching_only {
  forall Legal(Staff(name), TeachingOnly(_))
  in personnel.state {
    exists Staff(name, _, 0, _) in resource_planning.state
  }
}
    
```

Similarly, research active staff must be given at least 50% of their time for research:

```

research_active {
  forall Legal (Staff (name), ResearchActive (_))
  in personnel.state {
    exists Staff (name, _, research, _)
    in resource_planning.state { research >= 50 } } }

```

Finally, all research active staff should be given 3 contiguous days free of teaching where this is possible:

```

days_for_research {
  forall Legal (Staff (name), ResearchActive (_))
  in personnel.state {
    forall Booking (name, _, t)
    in room_booking.state {
      let week = [ time |
        Booking (name, _, time) <- room_bookings.state,
        ?(time >= t and time <= t+5) ]
      in if (length(week)) = 2
      then
        case bookings_for_week {
          [x,y] -> abs(x - y) = 1 or abs(x - y) = 4
        }
      else false } } }

```

The quality measure (Δ_i in F_2) is now redefined so that it is the *maximum* of the locally defined quality and the independently defined quality of the journal.

11 VERIFICATION

Section 8 defines the function and this section uses rigorous argument to establish the following proposition: Given the UoS architectures, C_a and C_b described in sections 6 and 10 respectively, then for any collection of simulation messages M for which $C_a \rightarrow_M T_a$ and $C_b \rightarrow_M T_b$, and under the assumption that both architectures use the same resources in which staff who are not research active can be identified, then $F(T_a) \leq F(T_b)$.

Proof: by case analysis on the components of F : (F_1) since C_b includes an invariant constraint that research active staff are given time to pursue research where possible, then the number of research outputs will be at least the same as in C_a ; (F_2) in C_a , Δ_i is defined in terms of a local measure for research output quality and in C_b , Δ_i is defined as the maximum of the local measure and the independent measure of journal quality, therefore the weighted sum of output quality will at least stay the same; (F_{3a}) the number of research active staff is the same as defined in the proposition above; (F_{3b}) the amount of time given to all staff to undertake research is opportunistic in C_a and is managed in C_b ; under the assumption that the manages process will identify contiguous time for research wherever this is possible then the time available to research active staff will be maintained or increase in C_b ; (F_4) the transfer from repository to web in C_a is opportunistic and is systematic in C_b , therefore the externally visible research profile of UoS will be maintained or increased in C_b compared to C_a . Therefore $F(T_a) \leq F(T_b)$ **QED**.

12 CONCLUSIONS

This paper has proposed an approach to EA alignment with respect to business goals that involves the use of precise architecture definitions that support goal measurement functions. We have reviewed the literature in this area and argued that existing technology for EA is not sufficiently precise to support such an approach. We have used the LEAP modelling technology to validate the approach with a real-world case study and have shown that precisely defined architecture models allow business goals to be represented as computable functions and therefore architecture models to be compared with respect to goal satisfaction.

We aim to take this approach further by integrating LEAP with formal methods technologies such as SAT solvers and model checkers in order to determine whether EA alignment and goal satisfaction modelling can be automated.

REFERENCES

- Avison, D., Jones, J., Powell, P., and Wilson, D. (2004). Using and validating the Strategic Alignment Model. *The Journal of Strategic Information Systems*, 13(3):223–246.
- Bergeron, F., Raymond, L., and Rivard, S. (2004). Ideal patterns of strategic alignment and business performance. *Information & Management*, 41(8):1003–1020.
- Chan, Y. and Reich, B. (2007). IT alignment: what have we learned? *Journal of Information Technology*, 22(4):297–315.
- Chung, S., Rainer Jr, R., and Lewis, B. (2003). The impact of information technology infrastructure flexibility on strategic alignment and applications implementation. *Communications of the Association for Information Systems*, 11(11):191–206.
- Clark, T. and Barn, B. (2011). Event driven architecture modelling and simulation. In *Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on*, pages 43–54.
- Clark, T. and Barn, B. S. (2012). A common basis for modelling service-oriented and event-driven architecture. In *Proceedings of the 5th India Software Engineering Conference, ISEC '12*, pages 23–32, New York, NY, USA. ACM.
- Clark, T., Barn, B. S., and Oussena, S. (2011). Leap: a precise lightweight framework for enterprise architecture. In *Proceedings of the 4th India Software Engineering Conference, ISEC '11*, pages 85–94, New York, NY, USA. ACM.
- Gregor, S., Hart, D., and Martin, N. (2007). Enterprise architectures: enablers of business strategy and is/it alignment in government. *Information Technology & People*, 20(2):96–120.

- Henderson, J. and Venkatraman, N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1):4–16.
- Luftman, J. (2004). Assessing business-IT alignment maturity. *Strategies for information technology governance*, page 99.
- Reich, B. and Benbasat, I. (1996). Measuring the linkage between business and information technology objectives. *MIS Quarterly*, pages 55–81.
- Spencer, J. et al. (2004). *TOGAF Enterprise Edition Version 8.1*.
- Street, C. and Denford, J. (2012). Punctuated equilibrium theory in is research. *Information Systems Theory*, pages 335–354.
- Tan, F. and Gallupe, R. (2006). Aligning business and information systems thinking: A cognitive approach. *Engineering Management, IEEE Transactions on*, 53(2):223–237.
- Vargas Chevez, N. (2010). A unified strategic business and IT alignment model: A study in the public universities of Nicaragua. In *AMCIS*. AIS Electronic Library, KTH.

