Goal based alignment of enterprise architectures

BARN, Balbir and CLARK, Tony <http://orcid.org/0000-0003-3167-0739>

Published version

BARN, Balbir and CLARK, Tony (2012). Goal based alignment of enterprise architectures. In: 7th International Conference on Software Paradigm Trends (ICSOFT 2012), Rome, Italy, 24-27 July 2012.

Goal Based Alignment of Enterprise Architectures

Balbir S. Barn, Tony Clark
School of Engineering and Information Sciences,
Middlesex University, London NW4 4BT, UK
{b.barn,t.n.clark}@mdx.ac.uk

April 19, 2012

Abstract

Business and IT alignment remains an ongoing concern for organizations. Enterprise Architecture has emerged as a possible tool for understanding and planning for BIA, but the current range of methods and frameworks for EA do not support precise measurement. 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. The goals enable description of business level requirements whilst the functions enable the goals to be evaluated at a technical architecture level. The technology is evaluated with a detailed and authentic case study from higher education.

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 and IT strategy [Chan and Reich, 2007]. While there are multiple definitions for business and IT alignment (BIA) reflecting the different foci including integration, linkage, bridge, fusion or even fit, we will utilize the definition derived from the *Strategic Alignment Model* (SAM) [Henderson and Venkatraman, 1993]. They state 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 essentials of a business, its IT and its evolution, and to support analysis of this information: '*it is a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems and infrastructure.*'[Lankhorst, 2009]. In addition to presenting a coherent explanation of the *what*, *why* and *how* of a business, EA aims to support specific types of business analysis including: alignment between business functions and IT systems; business change describing the current state of a business (as-is) and a desired state of a business (to-be). Thus EA has the potential to serve as the basis of machinery that can be used to address BIA [Wang et al., 2008, Pereira and Sousa, 2005].

Despite the opportunities presented for addressing 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 because of ambiguity between concepts, overlap of concepts and perhaps most importantly, no clear refinement relationships between business goals and enterprise architecture component changes. Some evidence of this latter issue can be noted in the Archimate language [Lankhorst et al.,] which is now beginning to include concepts such as goals as part of its meta model.

This paper contributes a rigorous approach to aligning business goals to IT infrastructure by proposing a method for describing business goals in terms of functions that can be applied to both as-is and to-be enterprise architectures. The approach is validated by the use of a method and supporting technology(simulation environment) against an authentic case study of required business change. Our fundamental assertion is that our method (LEAP or Light weight Enterprise Architecture Process) supports the encoding of business goals as functions that map to LEAP executions of both as-is and to-be EA.

The remainder of the paper is structured as follows: section 2 places our proposal for goal based EA alignment in the context of related work; section 3 describes our approach to using LEAP to modelling architectures, measuring goal satisfaction and alignment; section 4 describes a real-world case study that involves architectural modelling and a business goal that requires architecture modification; section 5 describes a collection of assumptions that leads to a specification of an as-is architecture in section 6; the semantics of LEAP are given in section 7 together with an evaluation function in section 8 that measures the degree to which the as-is architecture meets the case-study business goal; section 10 then modifies the architecture and in doing so claims to increase the business goal satisfaction; the claim is proved using rigorous argument based on LEAP semantics and the function definition in section 11.

2 RELATED WORK

Alignment between business and IT is an oft cited issue and challenge facing organizations. As a consequence, attempts to measure the extent of BIA within an organization has been extensively researched and some notable models for measuring BIA have included the leading model for alignment originally proposed by Henderson and Venkatraman, the Strategic Alignment Model (SAM) [Henderson and Venkatraman, 1993]. Other researchers have extended the SAM and it's associated definitions with varying success and usage. Luftman proposed the Strategic Alignment Maturity Model (SAMM) that 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]. Areas which should be the focus of future research include that recommended by Chung et al who suggest 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].

The measurement of alignment is important for several reasons. From a business perspective, a measurement or scale of alignment can lend itself to a managed alignment process. Researchers can use reliable and valid measures to support more rigorous research. And while there are a range of measures that have been utilised to date (see Chan et al [Chan and Reich, 2007] for a comprehensive list), measures that directly link business goals to applications and their implementation are lacking. Bergeron et al. also note that future research should adopt a dynamic rather than static perspective [Bergeron et al., 2001].

EA methods tend to either be isolated such that they take the situation of an application into account [Riege and Aier, 2009] or they tend to focus on architecture frameworks [Steen et al., 2004]. Frameworks provide a logical structure for classifying and organising representations of EA for specific stakeholders. Leading examples include: Zachman's Framework [Zachman, 1999]; The Reference Model for Open Distributed Processing (RM-ODP) and the Open Group's framework TOGAF and derivatives [Wisnosky and Vogel, 2004, MoD, 2008]. Regardless of the specifics of the framework, as Tang et al note there are common deficiencies such as: (1) the level of detail required in an architecture model is not generally specified; (2) support, specification and management of non-functional requirements is lacking and (3) software configuration modeling is also generally lacking [Tang et al., 2004].

Apart from the generic frameworks outlined above, there are EA methods that aim to address activities that support areas when architectural activity moves into detailed design. For example, Memo [Frank, 2002] is an EA method that introduces a range of visual modeling languages supporting multiple views. The method provides an integrated process model. Pereira and Sousa [Pereira and Sousa, 2004] introduce a method that is overlayed on top of the Zachman framework and suggests how specific techniques can be used to develop each of the 36 viewpoints. Wang et al present a high-level approach to EA [Wang et al., 2008]. Wegmann et al present SEAM (Systematic Enterprise Architecture Methods) - a family of methods that utilise elements of value chains and nested components to represent how business goals decompose to system level of components [Wegmann et al., 2007]. The SOMA method developed by Arsanjani et al for IBM is an end-end software development life-cycle method that assumes a service oriented architecture style for EA. The method uses concepts of component based design and goal oriented modeling as well as established techniques such as use case modeling to support the design and implementation of EA solutions [Arsanjani et al., 2008].

One of the emerging standardisation efforts on goal modelling is the Business Motivation Model (BMM) from the Object Management Group (OMG)[1]. This model provides a set of concepts for describing business plans and their elements together with how these elements interrelate. The BMM makes appropriate linkage to other reference models such as the Business Process Modelling Notation (BPMN). Our work utilises the Goal element from the BMM. Goal Modeling itself has been the subject of much research, especially in the context of requirements modelling, for example the i* model has been widely discussed. A useful overview of this area is described by [Kavakli and Loucopoulos, 2005].

This section has not attempted to provide an exhaustive review (paper size limitations prevents that), however it is noticeable that the current EA methods and frameworks do not link goals, EA and business alignment and there is limited research on how to formally verify the extent of the gap. This paper provides one such effort in this area.

3 LEAP GOAL MODELLING AND CONCEPTS

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 essence, the approach is presented as a domain specific language for EA description similar to architecture description languages, and is implemented as a Java based interpreter that drives a simulation environment that includes elements for visualisation of results. To support the approach described in this paper, we introduce additional concepts and method requirements that enable the technology to provide a simulation of the extent to which an as-is and to-be architecture is aligned with a business goal.

In order for an organization to understand the impact of a business goal both in terms of current technology 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 totally 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. Therefore, the architectures to be aligned (in this case as-is and to-be architectures) must be precisely captured as a model. Our claim is that any meaningful

[1] http://www.omg.org/spec/BMM/1.1/PDF/

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.

In principle, any technology can be used to implement this approach to alignment. This paper uses the LEAP language because it is simple and has a precise semantics. LEAP uses hierarchical components with ports and connectors to represent any system. Each component has a state that is a sequence of terms, invariants over the state, and a collection of operations that are invoked in response to receiving messages on input ports. An operation has a collection of specification clauses each consisting of a pre, post and message condition. For the clause to hold, and therefore define how the operation handles a corresponding message, the precondition must be true of the state, the postcondition must be true of the state before and after the operation handles the message, and the message condition must hold for all messages produced on the component's output ports. LEAP conditions are defined using boolean expressions that use both quantification and pattern matching in an intuitive way.

The semantics of LEAP models is given as a sequence of component states where each state is a set of terms. An evaluation function is defined to map a sequence of component states to an integer value that represents relative goal satisfaction. Modifications to component architecture models can then be compared by applying the evaluation function to their semantics and proving that the results are always increasing when the architectures start in equivalent states.

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. The UK HE landscape has also changed with the emergence of private providers and increasing distance and on-line education from international institutions all creating additional competition. 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]. Relatively, the academic domain has received less research focus on the study of the business of higher education and alignment with IS/IT. Sabherwal and Kirs investigated the alignment of critical success factors with IT capability and confirmed that alignment facilitates both perceived IT success and organizational performance (perceived IT success refers more specifically to the extent to which the senior managers of the organization believe IT capability contributes to the organization's success) [Sabherwal and Kirs, 1994].

To explore how technologies proposed in this paper can be utilised in the HE domain, we describe a real-life case study that captures the essence of the key challenge addressed at the beginning of this section. We introduce the case study here:

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 and the business objectives therein have introduced new requirements. Namely, the University will compete on quality and that quality will be exemplified by an increase in research outputs of higher quality. Such a change will hopefully lead to a higher ranking in the research league tables for Universities in the UK. These high level business goals will require changes to business processes, organisational culture and IT systems that will then deliver the business objectives.

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 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:

- Timetabling systems that schedule staff to specific modules to specific room.
- The workload management system used by Heads of Department to ensure that staff are allocated work in an appropriate manner.
- The University Repository that provides external access to the research outputs of the University.
- The HR system that manages contracts for staff.
- 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.

Both existing and the proposed systems need to include an ability to measure the "quality" or "extent" by which the two **as is** and **to be** configurations support the systems.

The remainder of this section demonstrates how the technology introduced earlier addresses the requirements emerging from the case study to provide a concrete case for measuring the gap between business and IT alignment.

5 Assumptions

The UoS business goal is to increase the visibility, quantity and quality of its research, and thereby increase its reputation. 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:

- In planning any changes to UoS no new staff resources can be made available. Therefore, it is not possible to improve reputation (in the short term) at UoS by poaching new staff.
- All changes must maintain the current levels of teaching.

UoS will need to change in order to increase its reputation. 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:

- Reputation will be enhanced by visibility of research outputs via the UoS web-site.
- UoS is effective at measuring the quality of its research outputs.
- Research is to be measured on a 4-point scale with 1 being the lowest level of quality. The likelihood of producing high-quality research is increased by providing staff with a contiguous period of time free from teaching and admin responsibilities.
- The number of research outputs and their quality are positively correlated to reputation.

6 Specification: As Is

The LEAP specification of the current UoS enterprise consists of a collection of components. The components define data models that reflect the current UoS support for and management of research. The 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 scrabbleshire {
  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. It is *logical* in the sense that there is no physical IT system in UoS that manages this information and therefore it is used to support the simulation:

```
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 to allocate members of staff to rooms at particular times throughout the teaching year. 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 scrabbleshire 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. A concrete LEAP specification allows components to be nested; the specification flattens this tree structure and allows child database elements to be referenced within parent databases (elements are assumed to be tagged). A concrete LEAP specification supports invariants on each component that are assumed to hold at all times within a component

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

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

$$\text{L-3}\;\frac{\begin{array}{c}M_1\vdash C_1\Rightarrow C_2,M_2\\ M_2\vdash C_2\Rightarrow C_3,M_3\end{array}}{M_1\vdash C_1\Rightarrow C_3,M_3}$$

Figure 1: LEAP Semantics

$$
\begin{array}{ll}
F_1(\_+[\_\cup\{(\mathtt{repository},\Delta)\}]) & = \#\Delta \\
F_2(\_+[\_\cup\{(\mathtt{repository},\Delta_1\cup\Delta_2\cup\Delta_3\cup\Delta_4)\}]) & = \sum_{i=1,4} \#\Delta_i * i \\
F_{3a}(+[\_\cup\{(\mathtt{personnel},\Delta_h),(\mathtt{repository},\Delta_r)\}]) & = \#\{n \mid \mathtt{Staff}(n)\in\Delta_h, \mathtt{Entry}(n,\_,\_)\in\Delta_r\} \\
F^s_{3b}(\tau_1:\tau_2:\tau_3:\tau_4:T) & = \begin{cases} 1+F^s_{3b}(T) & \textbf{when} \bigwedge_{i\in 1,4} \neg(\mathtt{teaching}(s,\tau_i) \\ F^s_{3b}(\tau_2:\tau_3:\tau_4:T) & \textbf{otherwise} \end{cases} \\
F^s_{3b}([]) & = 0 \\
F_{3b}(T) & = \sum_{s\in S} F^s_{3b}(T) \\
F_3(T) & = F_{3a}(T)+F_{3b}(T) \\
F_4(T+[\_\cup\{(\mathtt{web},\Delta)\}]) & = (\#\Delta)/(\#T+1) \\
F(T) & = F_1(T)+F_2(T)+F_3(T)+F_4(T)
\end{array}
$$

Figure 2: Goal Function

κ. Finally, a concrete LEAP specification supports ports and port connections. Ports are not represented in the semantics and port connections are handled by directing output messages to the target component.

An operation specification $m(\tilde{i})^{\mu}_{p,q} \in S$ 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 $\mathrm{FV}(p)$, 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 $M = \{\mathtt{Message}(t,n,a),\ldots\}$ and an execution trace $T = [C_0, C_1, \ldots, C_n]$ such that $[\mathtt{perform}()] \vdash C_0 \Rightarrow C_n, \emptyset$ and all messages M have been processed.

8 Goal Function

The UoS business goal can be implemented as a measure F on execution traces. The measure can be decomposed as follows:

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. This measure can be decomposed into:

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. # 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 $\texttt{Entry}(n,e,i,j)$. 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 $F(T)$.

9 Conformance

Our proposal is that an architecture can be expressed as a LEAP specification and a business goal can be encoded as a function that maps a LEAP execution to a value in a totally ordered set. Section 7 has defined how to map a LEAP specification to an execution represented as a sequence of component states. Section 8 has defined a function F that maps an execution trace to an integer in the context of the UoS business goal. 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 {
```

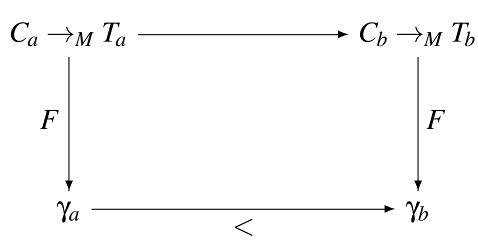

Figure 3: Goal Alignment

```
      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 UoS web site is updated to minitor the respository 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 wil 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

The UoS business goal is to increase reputation by maximizing the number, availability and quality of its research outputs. Our method is to convert the goal into a function over institutional behaviour and to show that changes in the UoS architecture increase the goal evaluation function. Section 8 defines the function and this section uses rigorous argument to establish the following proposition:

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. 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_3 Is decomposed into:

- 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 maximum number of web entries is the total number of repository entries. 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 CONCLUSION

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 technolgies such as SAT solvers and model checkers in order to determine whether EA alignment and goal satisfaction modelling can be automated.

References

[Arsanjani et al., 2008] Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., and Holley, K. (2008). Soma: A method for developing service-oriented solutions. *IBM systems Journal*, 47(3):377–396.

[Avison et al., 2004] 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 et al., 2001] Bergeron, F., Raymond, L., and Rivard, S. (2001). Fit in strategic information technology management research: an empirical comparison of perspectives. *Omega*, 29(2):125–142.

[Bergeron et al., 2004] Bergeron, F., Raymond, L., and Rivard, S. (2004). Ideal patterns of strategic alignment and business performance. *Information & Management*, 41(8):1003–1020.

[Chan and Reich, 2007] Chan, Y. and Reich, B. (2007). IT alignment: what have we learned? *Journal of Information Technology*, 22(4):297–315.

[Chung et al., 2003] 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 and Barn, 2011] 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 and Barn, 2012] 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 et al., 2011] 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.

[Frank, 2002] Frank, U. (2002). Multi-perspective enterprise modeling (memo) conceptual framework and modeling languages. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 1258–1267. IEEE.

[Gregor et al., 2007] 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 and Venkatraman, 1993] Henderson, J. and Venkatraman, N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1):4–16.

[Kavakli and Loucopoulos, 2005] Kavakli, E. and Loucopoulos, P. (2005). Goal modeling in requirements engineering: Analysis and critique of. *Modeling Methods and Methodologies*, page 102.

[Lankhorst, 2009] Lankhorst, M. (2009). Introduction to enterprise architecture. In *Enterprise Architecture at Work*, The Enterprise Engineering Series. Springer Berlin Heidelberg.

[Lankhorst et al.,] Lankhorst, M., Proper, H., and Jonkers, J. The Anatomy of the ArchiMate Language. *International Journal of Information System Modeling and Design*, 1(1).

[Luftman, 2004] Luftman, J. (2004). Assessing business-IT alignment maturity. *Strategies for information technology governance*, page 99.

[MoD, 2008] MoD, U. (2008). Ministry of defence architecture framework.

[Pereira and Sousa, 2004] Pereira, C. and Sousa, P. (2004). A method to define an enterprise architecture using the zachman framework. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1366–1371. ACM.

[Pereira and Sousa, 2005] Pereira, C. and Sousa, P. (2005). Enterprise architecture: business and IT alignment. In *Proceedings of the 2005 ACM symposium on Applied computing*, page 1345. ACM.

[Reich and Benbasat, 1996] Reich, B. and Benbasat, I. (1996). Measuring the linkage between business and information technology objectives. *MIS Quarterly*, pages 55–81.

[Riege and Aier, 2009] Riege, C. and Aier, S. (2009). A Contingency Approach to Enterprise Architecture Method Engineering. In *Service-Oriented Computing–ICSOC 2008 Workshops*. Springer.

[Sabherwal and Kirs, 1994] Sabherwal, R. and Kirs, P. (1994). The alignment between organizational critical success factors and information technology capability in academic institutions*. *Decision Sciences*, 25(2):301–330.

[Spencer et al., 2004] Spencer, J. et al. (2004). *TOGAF Enterprise Edition Version 8.1*.

[Steen et al., 2004] Steen, M., Akehurst, D., ter Doest, H., and Lankhorst, M. (2004). Supporting viewpoint-oriented enterprise architecture.

[Street and Denford, 2012] Street, C. and Denford, J. (2012). Punctuated equilibrium theory in is research. *Information Systems Theory*, pages 335–354.

[Tan and Gallupe, 2006] Tan, F. and Gallupe, R. (2006). Aligning business and information systems thinking: A cognitive approach. *Engineering Management, IEEE Transactions on*, 53(2):223–237.

[Tang et al., 2004] Tang, A., Han, J., and Chen, P. (2004). A comparative analysis of architecture frameworks. In *Software Engineering Conference, 2004. 11th Asia-Pacific*, pages 640–647. IEEE.

[Vargas Chevez, 2010] 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.

[Wang et al., 2008] Wang, X., Zhou, X., and Jiang, L. (2008). A method of business and it alignment based on enterprise architecture. In *Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on*, volume 1, pages 740–745. IEEE.

[Wegmann et al., 2007] Wegmann, A., Regev, G., Rychkova, I., Lê, L., De La Cruz, J., and Julia, P. (2007). Business and it alignment with seam for enterprise architecture. In *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*, pages 111–111. IEEE.

[Wisnosky and Vogel, 2004] Wisnosky, D. and Vogel, J. (2004). DoDAF Wizdom: A Practical Guide to Planning, Managing and Executing Projects to Build Enterprise Architectures Using the Department of Defense Architecture Framework (DoDAF).

[Zachman, 1999] Zachman, J. (1999). A framework for information systems architecture. *IBM Systems Journal*, 38(2/3).