ONLINE ANALYSIS OF ENGINEERING FRAMEWORKS FOR VARIOUS LIFE CYCLE STAGES

Bart H.M. GERRITSEN (*)
Bart.Gerritsen@planet.nl

(*) TNO Netherlands Organisation for Applied Scientific Research,
Wassenaarseweg 56, 2333 AL LEIDEN (The Netherlands).

Mots clefs :
Modélisation du plan (framework), caracteristiques de framework, data mining, recherche bibliographique, féderation des frameworks, framework crawler, ranger des frameworks
Keywords:
Framework formulation, framework properties, data mining, bibliographic analysis, federation of frameworks, framework crawler, ranking of frameworks

Abstract

Effective, adaptable and extendable frameworks can be regarded a key element for future sustainable whole-life holistic product approaches, like the development of intelligent products and the internet-of-things, agile manufacturing, smart product bundling, closed-loop life cycle management, asset management, etc. Frameworks also play a vital role in enterprise architecture and business organization. Peculiar enough and in contrast with model, technology and knowledge, framework properties form an oddly neglected engineering research field, despite the fact that frameworks are being proposed by the dozens. The question arises how to apply, reuse and extend all these frameworks in a well-formed manner so as to be able to verify, implement and build upon earlier results in a harsh industrial setting. Moreover, framework analysis, if somehow structured, can complement to standards, protocols, ontology's and other formalizations.

We carried out a comprehensive bibliographic study into frameworks and applied part-of-speech-based analysis on framework properties. To that extend, we mined the available meta data of thousands of engineering frameworks disclosed on the internet in the period after 2000 and developed methods and techniques to classify them according to indicative factors for goals, resources, application context, etc. To enable searching, matching, comparison and ranking, we explored measures indicative of similarity in (part of) the framework properties. More specifically, we calculated textual energy-based association strengths to determine cross-context framework applicability. This is believed to be a necessary first step towards the formation of federations of frameworks and contract-based transformation and frameworks. Results shown in this paper are promising, but a supporting (semi-)structure like a framework ontology, is believed to further raise the online search and analysis potential of our method.

1 Introduction

Data mining in Scopus shows that over the last decade of the more than 1.2 million engineering articles published in one of the 5000 peer-reviewed international engineering journals, way over a 1100 engineering articles bear the term *framework* in their title. This implies a rise from 2% up to 3% over the last decade. Furthermore, statistics show that a total in excess of 35000 articles contains the term *framework* either in their title, their keywords or their abstract (refer to Figure 1). These figures indicate that framework is thus besides an ordinary English word also a frequently used term in scientific literature, that somehow represents a *chunk of engineering research results*. But contrary to for instance model, ontology, or knowledge, its confined meaning in engineering has been worked to a much lesser extent: little can be found in literature about the properties of a framework, its historical developments, its description, its applicability, its extendibility, etc. Most of what can be found originates from IT research, allegedly somewhat ahead of engineering sciences.

With a fundament so faint and with so many engineering frameworks being proposed, the question arises as to what exactly this total scientific body of framework articles is representing. Can frameworks be (re)used, adapted, extended and effectively be applied in different contexts, for instance? After all it is more common to extend and combine existing frameworks to fit our purposes, rather than to define frameworks from scratch. What exactly determines the limits of framework applicability and how can these characteristics be traced back to the current framework descriptions? And furthermore: can frameworks be merged in federations (of frameworks) to obtain an accumulated (joint) effect? Similar question have been posed long ago with respect to the fusion of models, ontology's and (consolidated) knowledge (bases) but oddly enough not for frameworks.

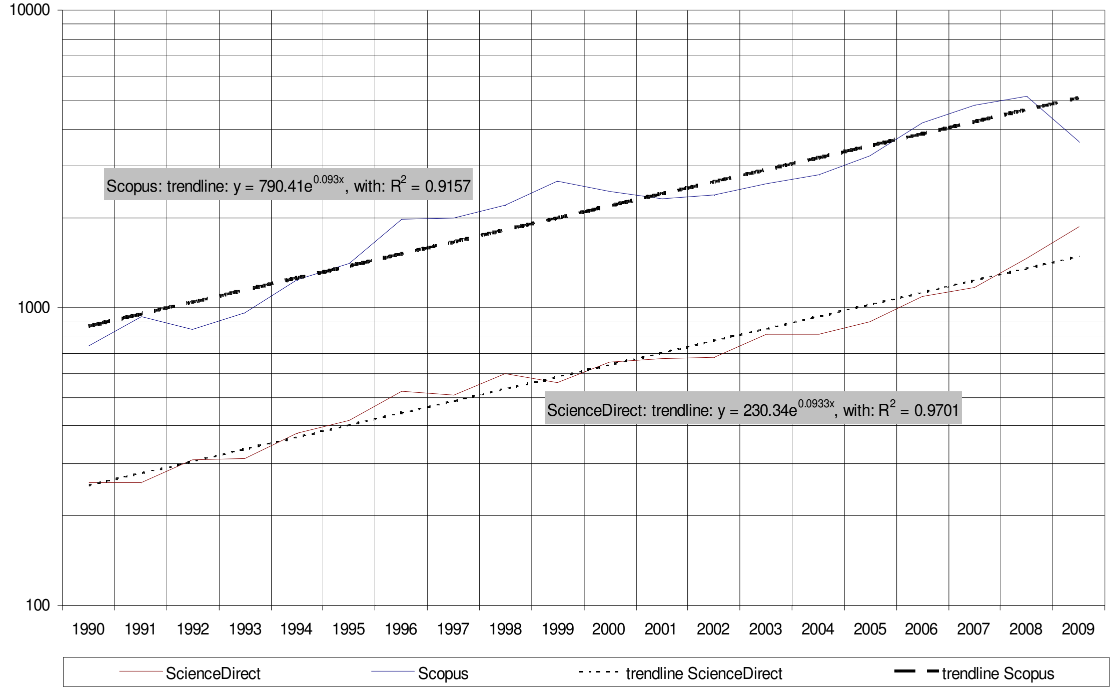

Figure 1: Full articles on frameworks over the last two decades in Scopus and ScienceDirect.

In an earlier paper [11] we showed that data mining and online analysis of framework paper titles, keywords and abstracts is an effective way to find, classify and select frameworks for further study and application. An online search and analysis tool suite, a *framework crawler*, has been proposed and explored, to serve as an enabling IT-technology that can be implemented to prune the internet for framework candidates. Using a framework crawler, one can mine structured and unstructured information on framework developments (such as scientific articles) and segment the results according to goal, technological age, application context, knowledge involved, etc.

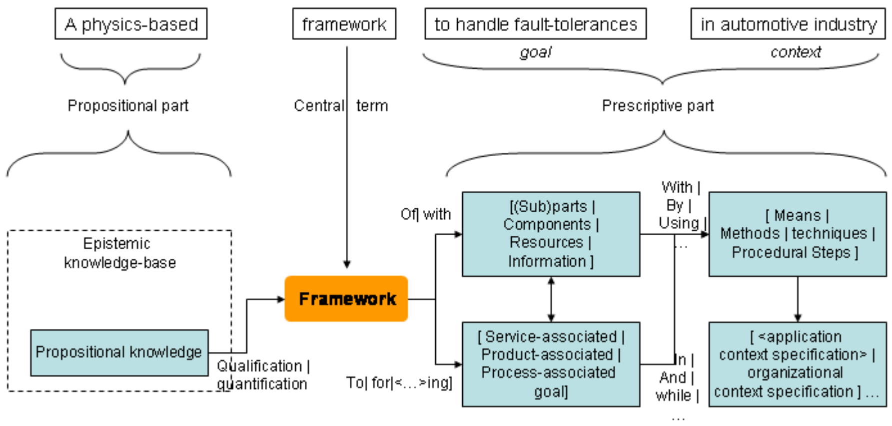

Figure 2: associative abstract framework model of [11].

Ideally, however, a combination of a formalized terminology and a (semi-) structured framework representation is used. This is believed to further raise the effectiveness of the classification of mined frameworks. A suitable such structure might be a framework ontology. Requirements and candidate frameworks on the internet can be matched by formulating and analyzing use cases and mapping requirements plus candidate frameworks onto such a structure. In [11], we developed an associative abstract framework model and part-of-speech-based analysis to classify frameworks by their bibliographic meta-data. A fitting framework ontology that pairs with this approach has not been seen in literature yet and may be developed later on. In this paper, building upon earlier results, we first concentrate on the matching process, however. We explore textual energy [7] as a measure to quantify a match between two or more frameworks and to rank a pool of candidate frameworks according to this measure. Such a measure can later on be included in a formal (semi-) structure. In the long haul, we seek to establish contract-based framework transformations that also allow for the assembly of federations of frameworks. A federation of frameworks cascades the effects of each of the single frameworks in the federation to reach a joint effect that cannot be obtained by a single framework alone. The output by one framework is used as (partial) input for the next. Here too, we need a measure to match in- and outputs.

Our main research objectives are the following:

- To make a first problem inventory;
- To set up an associative strength calculation to quantify similarity among frameworks;
- Idem, to quantify applicability of a framework, given a set of requirements or a use case model;
- To perform candidate ranking of framework;
- To explore new research avenues, towards contract-based framework transformations and federations of frameworks;

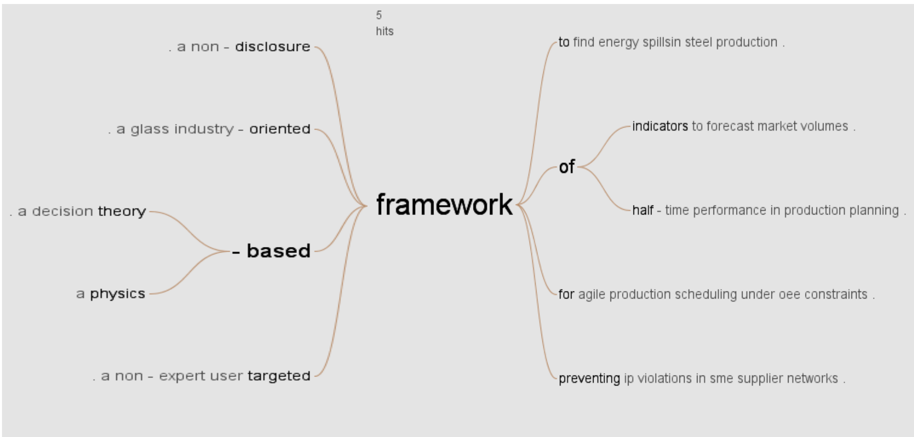

Figure 3: word chain illustrating the use of the associative model of Figure 2.

In this work, we restrict ourselves to engineering and engineering-and-IT frameworks. We also limit ourselves to English language peer-reviewed full journal articles published over the last decade. Furthermore, we limit ourselves to meta data as disclosed through Scopus and ScienceDirect in the form of bibliographic attributes, and do not necessarily enter into the full text of all these thousands of articles. We limit the research to four engineering topics; product development, product design, manufacturing, supply chain and only for specific comparison consider requirements specification frameworks too.

This paper is organized as follows. Firstly, in Section 2 we briefly overlook framework history and current state. Next, Section 3 discusses the main characteristics of frameworks as identified for this research. Then, in Section 4, we will elaborate upon the similarity measure for the comparison and ranking of frameworks. Section 5 shows the main results from this work, followed by concluding remarks in Section 6.

2 History and background

The origins of design and usage of frameworks is not well-marked in literature. One of the most active fields of multidisciplinary application-driven framework development is in MDO (multidisciplinary design optimization), in aeronautics and aerospace. Seminal work on frameworks formalization is due to Salas and Townsend [18], at NASA. They found historical time lines of MDO frameworks tracing back to the early nineties. Their 'major areas' in framework requirements covered architectural design, problem construction and execution and information access. Building upon this early work, further developments have been reported in [25][13][17][2][3] and [1]. Although highly optimized for the purposes, the relevance of these dedicated MDO frameworks is limited in view of the ambitions as in this paper: no results were reported with respect to similarity measures, framework transformation or framework federations. Mokyr [16] sets off *propositional* knowledge from *prescriptive* knowledge, to give foundation to the mapping of propositional *know-why* and *how-come* knowledge to prescriptive *know-how* and *how-to* knowledge in engineering applications. It is this notion that underpins the associative abstract model proposed in [11] and earlier work in [9]. Mokyr considers techniques, but his arguing can be generalized straightforwardly so as to also hold for frameworks. According to Mokyr, using techniques to create engineered artefacts does not require exhaustive knowledge of the underlying *epistemic* propositional knowledge, if staying within specific context-bounds for which the technique has been invented. To extend or *transform* an existing technique and to apply techniques beyond existing context, however, requires access to and understanding of the underlying principles and thus requires sufficient familiarity with the corresponding epistemic knowledge-base. This also holds for frameworks. Mokyr distinguishes between the *design* and the *usage* (community) of an engineering framework (technique). The history of linguistic and parts-of-speech analysis, and natural language processing in general, is well documented. See for instance [19], [15] or [8]. Chen in [4][5] discusses a tool-supported quantitative analysis method for POS-based research development visualization. Chen et al., in [6] considered textual energy and other measures for document-based co-citation quantification integrating various perspectives. Stumme in [20] proposed alternate knowledge visualization and mappings and recently, Lamar and Mocko [14] proposed the use of part-of-speech (POS)-based analysis techniques for the analysis of engineering requirement statements. Although capturing full semantics is beyond reach, they show that the more mechanical syntactic part of the analysis and verification of the bulk can be well treated using their approach.

3 Engineering frameworks

3.1 Definitions

For the further discussion of frameworks, we adopt the following definitions.

A *framework* is a set of conceptual ideas, practices and procedures, to achieve (a) technical predefined goal(s), given a set of resources, constraints and a modeled application context.

A *modeled application context* (or: *scope*), in this work, is a set of circumstances and characteristics describing the environment in which a framework is designed to be applied and operated.

The *actual application context,* in this work, is the whole of circumstances and characteristics that constitute the environment in which a framework is actually applied and operated.

A *federation of frameworks* is a constellation of two or more frameworks with the aim to establish a joint effect by putting (part of the) output by one framework as input to one of the other frameworks.

A *framework designer* is an expert who, whilst understanding and evaluating the propositional knowledge underpinning the framework validity and effectiveness, designs and documents a framework.

A *framework user* is an expert who, whilst understanding and knowing the actual context and the prescriptive knowledge about the framework under concern, evaluates whether this framework can be applied and operated validly and effectively in the actual context.

3.2 Framework elements

Framework aspects and elements [11];

- Have a (service-, product- or process-related or organizational) *goal* and generally, a framework description reflects procedural *ideas* of how to achieve that goal, using;
- Constituting sub-parts, *components*, resources or information;
- *Procedural (processing) steps* and their order, methods, techniques, etc;
- Bounded by an application *context* (application domain, conditions, technical constraints ...) and/or an organizational context (organizational level, audience, organizational constraints, …);

These elements appear in the form of *meta data* in bibliographic framework descriptions on the internet, such as titles, keywords, and abstracts. They can be mined using general data mining techniques and analyzed and classified as described in for instance [20],[4],[5],[6] and [11]. The above listed elements jointly make up the prescriptive part. Generally, framework descriptions also contain statements on the principles, theories, phenomena, etc. they are based upon. In our approach, these statements make up the propositional part of the framework description; see Figure 2.

3.3 Framework design

Like with product design, in practice, adapting and combining (parts of) existing designs is more common than designing a framework from scratch. Framework designers thus must be able to extract critical information from existing framework descriptions, to estimate the validity of modifications. To a larger extent, these *framework transformations* can be compared to model or web services transformations described in literature e.g., [12],[23].

3.4 Framework context

Frameworks exist in virtually any application domain: finance, economics, social sciences, life sciences, but also physical sciences and in the engineering domain. Sometimes, a single framework is applied across multiple such domains. When discussing the context (or: scope) of a framework, it is important t distinguish between the various types of context, such as application context, organization context and control context. In this work, we adopt the Zachman Framework [24] to denote and frame the context of frameworks under study. To study frameworks on product life cycle aspects, an auxiliary life cycle model may be merged.

It must be said that the term Zachman Framework may be a bit confusing; we try to avoid embarking upon a hierarchy of frameworks, but in this case the term Zachman Framework cannot be avoided and actually serves as a global architectural framework at the level of the organization as a whole. Zachman integrates principal views from principal viewpoints using distinct models (Figure 4). We try to associate framework contexts with such a principal view (e.g. a designers' view) and a viewpoint (e.g. how) to identify the context of a framework under concern.

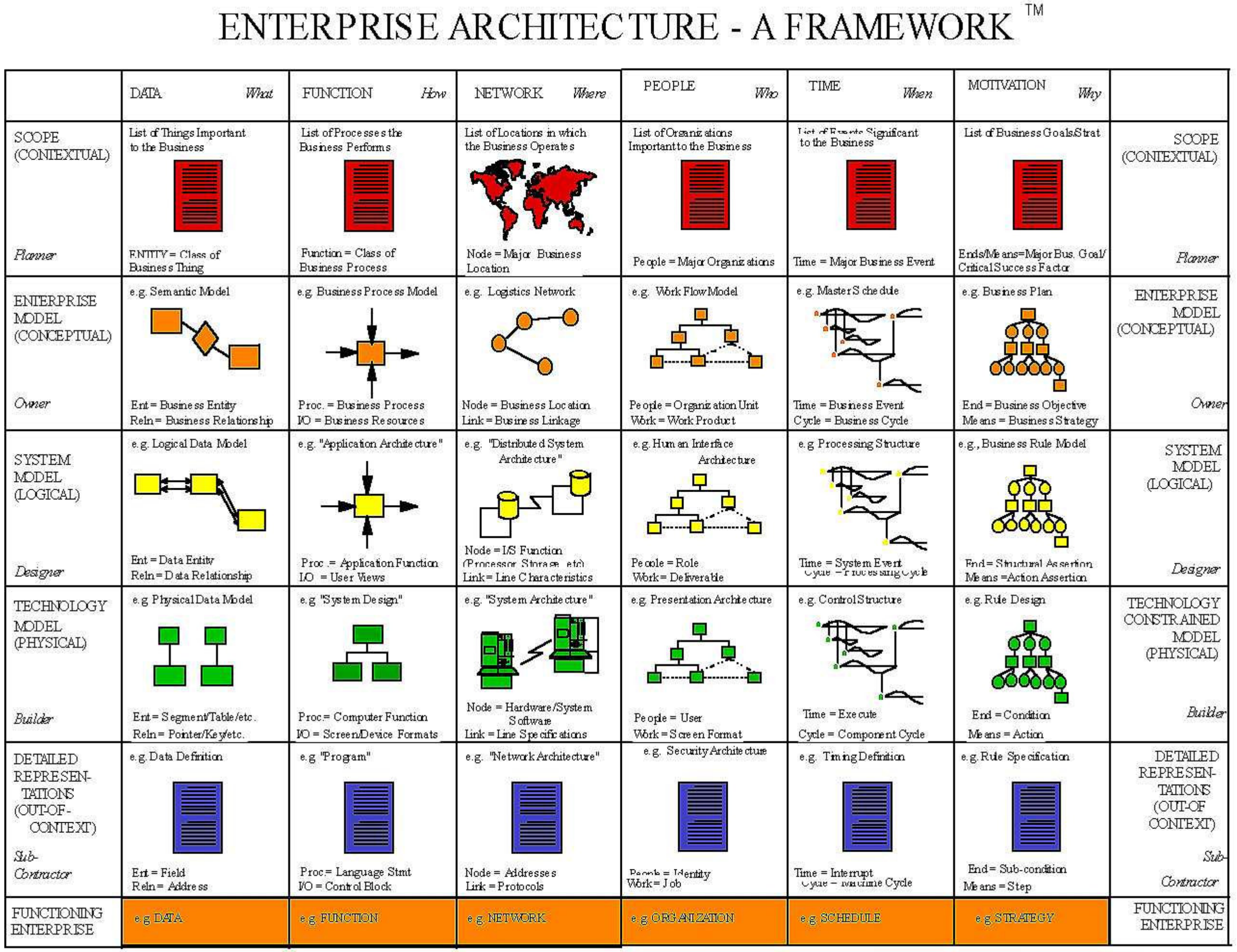

Figure 4: Zachman Framework with views row-wise and viewpoints column-wise. Source: http://www.zachmaninternational.us

3.5 Framework analysis

To analyze mined framework descriptions, we use POS-based analysis techniques. This means, we do not just analyze individual words but apply language rules (part-of-speech) to identify for instance goal specifying parts in article titles. We do this by applying syntactic sentence construction patterns: we search for the word ‘to’ expecting a goal description to follow. The same for words like ‘for’ (goal or purpose), ‘using’ (resources), ‘in’ (context), and a number of other *pivoting* words. Table 1 shows few such pattern templates.

Table 1: POS-pattern templates

A <some level>	framework	to	<achieve some goal>	using <some resource>	in <some context>
A <…>-based	framework	for	<some goal or purpose>	using <some resource>	in <some context>
A <…>	framework	of	<some component or resource>	to <achieve some goal>	in <some context>

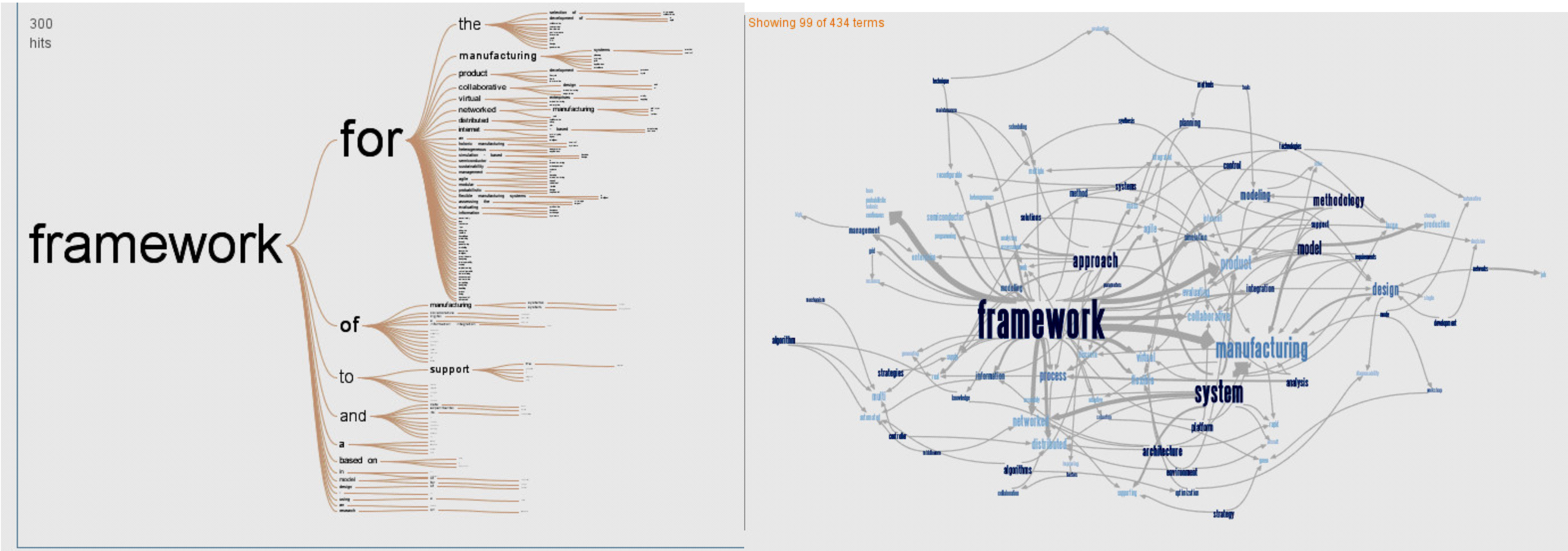

Figure 5: framework analysis results. Left : word chain starting with the word ‘Framework’. The bigger the font, the more occurrences found. Right : phrasenet diagram for the word ‘for’ : dark and light terms connected by a grey arrow form a phrase. Dark terms appear immediately before the word ‘for’, light terms immediately after. The phrase ‘framework for manufacturing’ stands out in the diagram. The thicker the grey arrow, the more occurrences found.

We developed a suite of Perl and AWK scripts for the analysis and use CiteSpaceII [4][5][21] and IBM/AlphaWork ManyEyes for visualizations (e.g., Figure 5). Statistical analysis on a study data set enabled us to derive a Markov Chain model with which we can estimate the likelihood of for instance an application in automotive, for a framework on product data exchange using STEP, by quantifying the likelihood of chain of words in article titles. The occurrence of a chain of words like: 'A framework to share and exchange product data in the automotive industry' can thus be predicted. This is an important capacity in view of our attempt to establish federations of frameworks using data mining and a (semi-)structure to measure.

3.6 Framework similarity measure

We use per Zachman level (scope level, enterprise (organizational) level, system level, technology-level …) and per Zachman viewpoint (what, how, where …) textual energy similarity measures as described in [7] to calculate framework similarities. An alternate method to calculate similarity measures is the n-gram based phrasenet (Figure 5, right

$d(t_c,t)=-3$			$d(t_c,t)=-2$			$d(t_c,t)=-1$			$t_c = CT$	$d(t_c,t)=1$			$d(t_c,t)=2$			$d(t_c,t)=3$			d(t
Term	freq	%	Term	freq	%	Term	freq	%		Term	freq	%	Term	freq	%	Term	freq	%	Term
	13	6.2%	A	27	12.8%	A	34	16.1%	F	for	100	47.4%	the	16	7.6%	design	19	9.0%	of
	7	3.3%	An	10	4.7%	modeling	8	3.8%	F	to	19	9.0%	product	12	5.7%	product	8	3.8%	and
	5	2.4%	and	7	3.3%	integration	8	3.8%	F	of	17	8.1%	support	11	5.2%	and	6	2.8%	design
	5	2.4%	product	7	3.3%	a	7	3.3%	F	and	13	6.2%	on	5	2.4%	the	5	2.4%	information
ign	5	2.4%	an	5	2.4%	design	7	3.3%	F	based	7	3.3%	collaborative	5	2.4%	of	5	2.4%	development
	4	1.9%	design	5	2.4%	modelling	5	2.4%	F	with	4	1.9%	concurrent	3	1.4%	integration	5	2.4%	information
	3	1.4%	the	4	1.9%	system	5	2.4%	F	in	3	1.4%	Web-based	3	1.4%	development	4	1.9%	for
em	3	1.4%	information	4	1.9%	conceptual	5	2.4%	F	model	2	0.9%	evaluation	3	1.4%	a	4	1.9%	product
	3	1.4%	a	4	1.9%	integrated	5	2.4%	F	and	2	0.9%	for	3	1.4%	management	3	1.4%	platform
eric	2	0.9%	-	3	1.4%	theoretical	4	1.9%	F	supporting	2	0.9%	development	2	0.9%	to	2	0.9%	engineering
l	2	0.9%	of	3	1.4%	engineering	4	1.9%	F				to	2	0.9%	function	2	0.9%	concurrent
dy	2	0.9%	on	3	1.4%	software	4	1.9%	F				integrating	2	0.9%	support	2	0.9%	knowledge
duct	2	0.9%	data	3	1.4%	and	3	1.4%	F				its	2	0.9%	communication	2	0.9%	sharing
ributed	2	0.9%	function	2	0.9%	the	3	1.4%	F				virtual	2	0.9%	tool	2	0.9%	integration
nufacturing	2	0.9%	via	2	0.9%	recovery	3	1.4%	F				optimal	2	0.9%	distributed	2	0.9%	data
	2	0.9%	processes	2	0.9%	unified	3	1.4%	F				knowledge	2	0.9%	planning	2	0.9%	environment
			knowledge	2	0.9%	optimization	3	1.4%	F				new	2	0.9%	knowledge	2	0.9%	management
			Part	2	0.9%	management	3	1.4%	F				integrate	2	0.9%	engineering	2	0.9%	
			Research	2	0.9%	development	2	0.9%	F				design	2	0.9%	conceptual	2	0.9%	
			cost	2	0.9%	generic	2	0.9%	F							lifecycle	2	0.9%	
			models	2	0.9%	decision-making	2	0.9%	F							rapid	2	0.9%	
			design-decision	2	0.9%	making	2	0.9%	F							collaborative	2	0.9%	
			decision	2	0.9%	support	2	0.9%	F							products	2	0.9%	
						information	2	0.9%	F							enterprises	2	0.9%	
						on	2	0.9%	F										
						Internet-based	2	0.9%	F										
						component	2	0.9%	F										
						reference	2	0.9%	F										
						exchange	2	0.9%	F										
						collaborative	2	0.9%	F										
						general	2	0.9%	F										
									211										
SING	79	37.4%	MISSING	16	7.6%	MISSING	0	0.0%	-	MISSING	33	15.6%	MISSING	33	15.6%	MISSING	38	18.0%	MISSING

Figure 6: frequency analysis on a study data set to derive Markov Chain likelihoods. The seven column table captures the linguistic environment centered around the central term 'Framework' denoted in the central column by the capital 'F'. The distance 'd(t_c,t)' is the relative position 'away' from this central term CT=t_c.

diagram) and quantitative word chains (Figure 5, left diagram and also Figure 7). Figure 7 shows the organizational and application context of frameworks in manufacturing and this method can generally be used to compare contexts. The following example explains how.

Consider two framework article titles:

Framework 1: ‘An education and experience-based framework to organize new product development teams in virtual manufacturing’;
Framework 2: ‘How to organize product development teams based on education and experience in a networked manufacturing environment? A framework ‘;

Firstly we rephrase framework 2 (refer to [11] for details) as follows: ‘An education and experience-based framework to organize product development teams in a networked manufacturing environment’, to make both framework 1 and 2 fitting the common pattern: ‘A <..> framework to <some goal> in <some application scope>’ as explained earlier. Here, the goal is to organize product development teams (taking into account education and experience) and the application scope is virtual manufacturing for Framework 1 and networked manufacturing for Framework 2. These two application scopes can be found in Figure 7. While the goal is more or less similar (how similar will be discussed later on), the application scope differs. Both frameworks may be paired to study cross-applicability (applying one in the application scope of the other) and/or studied for federation forming. The latter is interesting in case the frameworks take into account complementary Zachman viewpoints: if for instance Framework 1 would focus on ‘how to’ organize such teams (functional/procedural) and Framework 2 focuses on ‘who’, ‘why’ and ‘where’. A phrasenet diagram Figure 5, right part) and a Markov Chain is used to identify and quantify the (expected) occurrence of similar phrases in the frameworks to match. For instance, Framework 1 and 2 (after re-phrasing) share phrases like ‘education and experience-based’ and ‘organize product development teams’. These phrases can be entered in a phrasenet and following graph paths in the phrasenets quantifies the immediate linguistic environment of the constituent terms. The Markov Chain uses the observed frequency to quantify expected frequencies. In [11] we also used classical keyword-in-context (KWIC) computations to quantify observed frequencies.

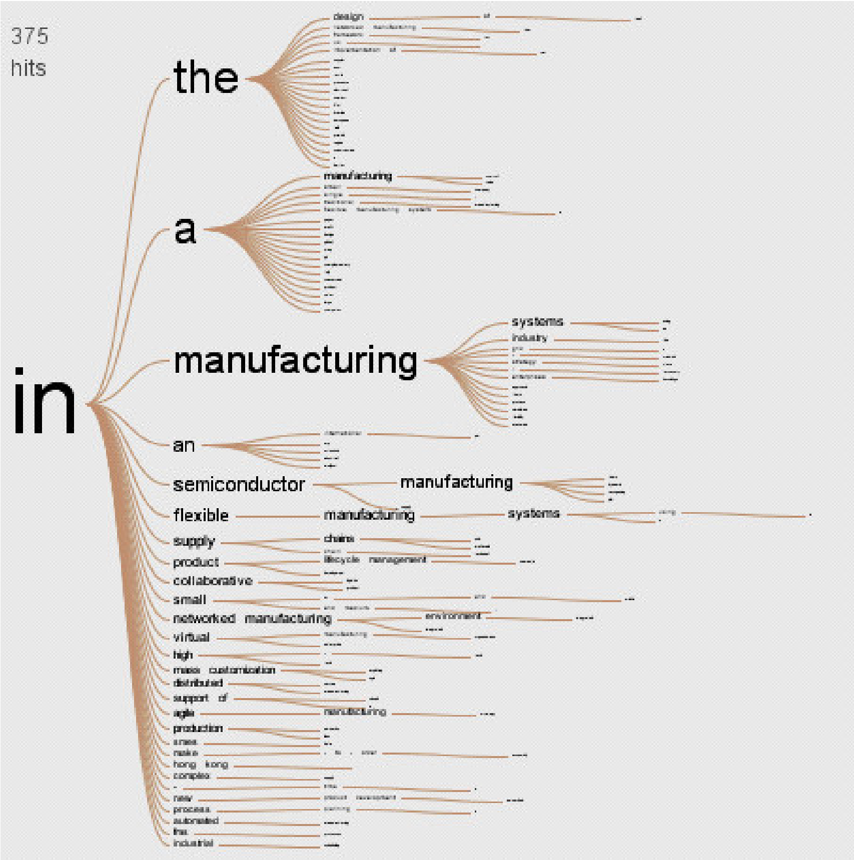

Figure 7: scope analysis; the taller the term, the more freqently occurring.

All these methods are used to match one part of the framework description (for instance: the goal part) while varying the other part that differs (for instance: the application scope). The challenge is then to measure (to predict) the *transformability* of the framework. Transformability is defined as follows. Given a current framework application utility or performance Π and an application scope Σ and an objective framework application scope Σ', the transformability is a measure for the framework capacity to achieve application performance Π in the objective framework application scope Σ'. We will say that the framework is transformed using transformation T:F(Π,Σ)→F(Π,Σ') . The aim is to ultimately arrive at a formalism for contract-based framework transformation, in similar vein to the notion of contract-based model transformation [12][23]. This allows for extensions, reuse and recombination of existing frameworks in new application scopes in a controlled and (semi-)structured manner. In the following section, we discuss a measure to assess and quantify the match between the common framework description part (like the goal part) and the difference (the 'distance' ΔΣ=Σ'-Σ to be abridged in the transformation) in the varied part (like the application scope). One final remark applies: in the examples given here, we match by goal and vary the application scope. This is not the only possibility. We can also compare by a common scope (like: the semiconductor industry, or: agile manufacturing, or: conceptual design, or: virtual enterprising) and vary the framework goal part. This is basically what we do when seeking for framework federations. When integrating a federation on viewpoints, we seek to match along Zachman rows (Figure 4), when integrating levels we seek to match along Zachman columns.

4 Measure of similarity

In this section, we will further work out the measure based on the notion of textual energy introduced in Section 3.6. Consider a propositional knowledge domain Ω (of 'how come' and 'why is'-knowledge; [9]) in which the full epistomological knowledge base for framework design is contained, and a prescriptive knowledge domain λ (of 'how to'-knowledge). Conforming to the associative model in Figure 2, we describe a framework mapping F as:

$$F : \mathrm{Z} \rightarrow \hat{\lambda} \qquad (1)$$

where:

F	=	mapping by the framework
$\mathrm{Z} \subset \Omega$	=	Epistemic knowledge-base in Ω
$\hat{\lambda} \subset \lambda$	=	Prescriptive co-domain

Not all useful knowledge in Ω is needed and used by a framework, so we assume a proper subset and assume mapping F a surjection. Furthermore, assume an alphabet Α of characters with which we can form words and phrases, we generally denote by the English word *term t*. For now, characters themselves, grammar and any other further linguistic parameters remain further unspecified and are simply assumed identical to international common journal language US English. With t_c being a *central term*, in our case t_c='framework', a linguistic environment $LE(t_c,s)$ centered on t_c with span s is defined as:

$$LE(t_c, s) = \left\langle t_i \ldots t_{-1}, t_c, t_1, \ldots, t_j \right\rangle \qquad (2)$$

where:

t_i, t_j	=	Term at the *i*-th position left, resp. *j*-th position right of central term t_c
$\langle ... \rangle$	=	(semi)-ordered list of terms

with span $s = [i, j],\ i \le 0 \le j$. Further, define $\|s\|$ the length of *LE*, equal to $\|LE(t_c, s)\| = \|\langle t_i ... t_j \rangle\| = |\,i\,| + 1 + |\,j\,|$ terms. In case $i = -j$ this reduces to $\|s\| = 2j + 1$. Observe that *s* is basically an interval of terms, centered at central term t_c. We rewrite form (2) so as to be able to account for Ω-domain related terms and for the prescriptive terms associated with the λ-domain. We will conveniently call this the Ω-list and the λ-list:

$$LE(t_c, s) = \langle LE^{-}_{Z} \mid t_c \mid LE^{+}_{\lambda} \rangle \qquad (3)$$

where notation:

$\langle L_1 \mid L_2 \mid L_3 \rangle$	=	A concatenation of lists $L_1 ... L_3$

which only holds iff:

$$\begin{array}{ll} (i) & LE^{-}_{Z}(t_c, s_i) = \langle t_i .. t_{-1} \rangle \\ (ii) & LE^{+}_{\lambda}(t_c, s_j) = \langle t_1 .. t_j \rangle \end{array} \qquad (4)$$

and:

$s_{i,j}$	=	The i- resp. j-part of span *s*

Using set $T = T_Z \cup T_\lambda$, with $T_Z = \{t_i\}$ and $T_\lambda = \{t_j\}$ and $LE = t_c \cup LE_T = t_c \cup T$, this is equivalent to condition:

$$T_z \cap T_\lambda = \varnothing \qquad (5)$$

which will not always be the case in practice, but can often be reached by rewriting the linguistic environment without changing its semantic meaning;

- Firstly, condition (5) is often violated by (English language) articles, which can be easily remedied by filtering them off;
- Condition (5) violations can also be circumvented by inserting thesaurus-based synonym substitutions;
- Finally, rotation techniques like in a KWIC procedure can assist in rephrasing unaligned forms into aligned forms.

An example taken from [11] may illustrate this (see: Example). In practice, far more intricate problems may occur, for which more advanced natural language processing techniques are needed. We are not going further into that in this research.

> *Example*
>
> Title of article:
>
> *Moving beyond the current technical limitations of personal computer-based FEM: a CAE-based framework.*
>
> which is of the form:
>
> $$LE(t_c, s) = \langle LE_T \mid t_c \mid \varnothing \rangle$$
>
> but can be rewritten as follows:
>
> *A CAE-based framework to move beyond the current technical limitations of personal computer-based FEM.*
>
> which after rewriting, takes the form:
>
> $$LE(t_c, s) = \langle LE_Z^- \mid t_c \mid LE_\lambda^+ \rangle$$
>
> with:
>
> $$LE_\lambda^+ = \{to, move, ..., FEM\}$$

Computing similarity among two framework description is a two-step process: step 1 is a match of each of the frameworks by a set of target terms, step 2: is an interaction and energy computation given the matches of step 1. Let us start looking at step 1 first. A common article is generally meaningless, in other words: we are interested in a match by *specific engineering terms*, such as product, development, design,materials,…) only. Following [7], we therefore define a set of *target terms* W by which we will seek to match each framework in step 1. Given a framework 1 and a framework 2 with a linguistic environment (Figure 2) LE_1 and LE_2, resp. a match is trivially obtained if we have that $LE \cap W \neq \varnothing$, i.e., if an element (a target term) $w_k \in W$ occurs in *LE*. Cardinality $|LE \cap w_k, w_k \in W|$ is a measure for the *potential* strength of the match. We can only quantify the actual strength in step 2. How do we compare two framework descriptions LE_1 and LE_2, or a single framework description LE_1 with a set of candidate framework descriptions $\{LE_j^*\}$, in step 2? Given the matrix S from step 1 with a match s_{jk} of each of the target terms $w_k \in W$, for each framework LE_j^*:

$$s_{jk} = |LE_j^* \cap w_k| \tag{6}$$

With $|W| = K$ target words and $|\{LE_j^*\}| = Q$ frameworks, we thus obtain a $Q \times K$ matrix S. $K \times K$ matrix $S^T S = J$ is called the *interaction matrix* between the matching terms and the *textual energy TE* is now defined as follows:

$$TE = -\frac{1}{2} M = -\frac{1}{2} SJS^T \tag{7}$$

$Q \times Q$ matrix $M = SJS^T$ is a factorable positive definite symmetric matrix relating any two frameworks by their shared target terms $w_k \in W$. By taking this measure, direct sharing of terms lead to a non-zero *bond*. But also indirect sharing, via a shared third framework (called: a shared *environment*), leads to a bond as we will demonstrate below. If $\det M = 0$, the system contains a candidate framework LE_j^* that does not contain any of the target terms. This framework can be removed from the set of candidates without modifying any other match. Matrix M has furthermore the property that $\forall_{i,j} : m_{ij} \geq 0$ and $\forall_{i \neq j} : m_{ij} \leq m_{ii}$ and therefore if $\forall_{i,j \neq i} : m_{ii} = 0 \rightarrow m_{ij} = 0$. So if $trace\ M = 0$, none of the candidates contains any of the target terms $w_k \in W$ and the textual energy among the frameworks is zero: there is no relationship; the frameworks are orthogonal for that target term vector. For ranking, the absolute value of the similarity is not always relevant and therefore we may want to normalize matrix M which is more or less trivial: $\hat{M} = M / m_{\max}$. By here, we know that $| m_{\max} | = m_{\max} \geq 0$ and is found on the diagonal of M. Ranking according to the values in $\hat{M}$ leads to a semi-ordered list, because $m_{\max}$ is generally not unique. A few concluding remarks: We may want to restrict matching to one part of the framework description, while varying the remaining part, as explained earlier. We also may want to augment terms with synonyms from a thesaurus. To measure dissimilarity, we may want to match antonyms. We haven't worked this out so far, however. Finally, the above similarity measure is a self-consistency in the sense of [22].

Assume now a non-zero similarity of one part of the framework description. The probability of finding a framework of which the remaining part differs at most by a certain dissimilarity or distance $d(LE_{\lambda_2}^+, LE_{\lambda_1}^+) < \varepsilon$ is given by $\Pr\left[LE_{\lambda_2}^+ \mid LE_Z^- \wedge LE_{\lambda_1}^+, t_c, d(LE_{\lambda_2}^+, LE_{\lambda_1}^+) < \varepsilon\right]$.

5 Results

In this section, we demonstrate the use of association-based similarity measures to compare frameworks. To study and verify the behavior of our approach, we mined full articles from Scopus on framework for the application scope manufacturing, product design, product development and supply chains, in the period after 2000. Apart from the first example sets, all frameworks (of which only titles are given) can be found in the Scopus database. Given the first set of frameworks below and a target term vector *W=<knowledge,portfolio,aerospace industry* >. Target term matches (step 1) are bold and underlined.

j	Framework description	Knowledge	Portfolio	Aerospace industry
1	A strategic marketing <u>**knowledge**</u>-based framework for supply chain organization in the <u>**aerospace industry**</u>	1	0	1
2	A <u>**knowledge**</u> <u>**portfolio**</u>-driven framework for product prospecting	1	1	0
3	A <u>**portfolio**</u>-based customer classification framework for the <u>**aerospace industry**</u>	0	1	1

The resulting matrix $\hat{M}$ (step 2) is then as follows:

```
       1      2      3
   ------------------
1|  1.00   0.83   0.83
2|         1.00   0.83
3|                1.00
```

Observations:

- Neither the determinant nor the trace is zero, so there is a non-zero similarity and all frameworks participate in this similarity;
- As a result the matrix is full (only the upper triangular part is shown, the rest is found by symmetry);
- Each framework description contains exactly 2 target terms and each pair of frameworks shares a target term;

What happens is we drop a bond? We amend framework description LE_3 by removing the 'in the aerospace industry' application scope. We thus obtain: $LE_1 \cap LE_2 = \{knowledge\}$, $LE_1 \cap LE_3 = \varnothing$ and $LE_2 \cap LE_3 = \{portfolio\}$. The table below reflects this.

j	Framework description	Knowledge	Portfolio	Aerospace industry
1	A strategic marketing **knowledge**-based framework for supply chain organization in the **aerospace industry**	1	0	1
2	A **knowledge** **portfolio**-driven framework for product prospecting	1	1	0
3	A **portfolio**-based customer classification framework	0	1	0

Target term vector *W*=<*knowledge,portfolio,aerospace industry* > remains identical. The resulting matrix $\hat{M}$ is now as follows:

```
       1      2      3
   ------------------
1|  0.83   0.67   0.17
2|         1.00   0.50
3|                0.33
```

The result has changed. The phenomenon to observe here is that although $LE_1 \cap LE_3 = \varnothing$, $m_{13} \neq 0$. The reason for that is that target term *knowledge* connects LE_1 and LE_2 and LE_3 was already connected to LE_2 by the term *portfolio*. Now LE_2 serves as a shared *environment* of LE_3 and LE_1. As a result $m_{13} \neq 0$. Observe a drop, though, from 0.83 down to 0.17 in the matrix. Removing target term *aerospace industry* does not affect matrix $\hat{M}$.

Removing the connecting term *knowledge* disconnects shared environment LE_2 from LE_1 and consequently, the indirect relationship will cease to exist: m_{13} becomes 0. Further to this phenomenon, we expand the set of framework descriptions as follows.

j	Framework description	Knowledge	Portfolio	Customer
1	A strategic marketing **knowledge**-based framework for supply chain organization in the aerospace industry	1	0	0
2	A **knowledge** **portfolio**-driven framework for product prospecting	1	1	0
3	A **portfolio**-based **customer** classification framework	0	1	1
4	A cultural preferences-based framework for the design of e-**customer** portals	0	0	1

Let *W*=<*knowledge,portfolio,customer*>. Although 3 intersections are empty, the matrix remains full, except for framework 1 with framework 4; Indeed, LE_4 and LE_1 have neither a connecting target term, nor a shared environment.

```
        1      2      3      4
    ---------------------------
 1|   0.33   0.50   0.17   0.00
 2|          1.00   0.67   0.17
 3|                 1.00   0.50
 4|                        0.33
```

Reconsider the same set of frameworks. When no target term occurs, we obtain a zero-matrix $\hat{M}$ (below, left matrix). Observe that determinant and trace are both zero for this $\hat{M}$. If a single target term is matched (middle matrix) or even all target terms, *but none shared*, there is no similarity as expected.

```
        1      2      3      4
    ---------------------------
 1|   0.00   0.00   0.00   0.00
 2|          0.00   0.00   0.00
 3|                 0.00   0.00
 4|                        0.00
```

```
        1      2      3      4
    ---------------------------
 1|   1.00   0.00   0.00   0.00
 2|          0.00   0.00   0.00
 3|                 0.00   0.00
 4|                        0.00
```

```
        1      2      3      4
    ---------------------------
 1|   1.00   0.00   0.00   0.00
 2|          1.00   0.00   0.00
 3|                 1.00   0.00
 4|                        1.00
```

Next, consider a use case model in which the need for frameworks is specified. We demonstrate how candidate frameworks can be matched using our approach. We selected frameworks fitting the pattern: ‘A framework to ….’ The below table sets up the problem:

USE CASE	A framework to organize customer order intake and progress communication
CANDIDATE	A decision support system framework to process customer order enquiries in SMEs
CANDIDATE	Using axiomatic design with the design recovery framework to provide a platform for subsequent design modifications
CANDIDATE	A new conceptual framework to improve the application of occupational health and safety management systems
CANDIDATE	An integrated modelling framework to support manufacturing system diagnosis for continuous improvement
CANDIDATE	A tolerancing framework to support geometric specifications traceability
CANDIDATE	A generic framework to support the selection of an RFID-based control system with application to the MRO activities of an aircraft engine manufacturer
CANDIDATE	Internet-based framework to support integration of the customer in the design of customizable products
CANDIDATE	A formal framework to integrate express data models in an extended enterprise context
CANDIDATE	An information-integrated framework to support e-Manufacturing
CANDIDATE	Fuzzy multiple objective programming framework to prioritize design requirements in quality function deployment
CANDIDATE	The creation of output and quality in services: A framework to analyze information technology-worker systems
CANDIDATE	A framework to support customer-company interaction in mass customization environments
CANDIDATE	Developing a PDM/MRP integration framework to evaluate the influence of engineering change on inventory scrap cost
CANDIDATE	Recognizing features from engineering drawings without using hidden lines: A framework to link feature recognition and inspection systems
CANDIDATE	Intelligent agent framework to determine the optimal conflict-free path for an automated guided vehicles system
CANDIDATE	A framework to develop an enterprise information portal for contract manufacturing
CANDIDATE	Developing an integration framework to support the information flow between PDM and MRP
CANDIDATE	An information modelling framework to support intelligent concurrent design and manufacturing of sheet metal parts
CANDIDATE	The application of UML and an open distributed process framework to information system design
CANDIDATE	TAPAS: A modular framework to support reuse in scheduling software development
CANDIDATE	WeBid: A Web-based framework to support early supplier involvement in new product development

Use case description and candidates are merged in a common problem space. We take $W=\langle customer,order,progress \rangle$ in conformance with the use case. We omit the varying part (the application context) from the use case description in the system. This part may vary. The above system generated the following matrix $\hat{M}$:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	1.00	0.80	0.00	0.00	0.00	0.00	0.00	0.47	0.00	0.00	0.00	0.00	0.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2		0.67	0.00	0.00	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3			0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5					0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6						0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7							0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8								0.27	0.00	0.00	0.00	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9									0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10										0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11											0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12												0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13													0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14														0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15															0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16																0.00	0.00	0.00	0.00	0.00	0.00	0.00
17																	0.00	0.00	0.00	0.00	0.00	0.00
18																		0.00	0.00	0.00	0.00	0.00
19																			0.00	0.00	0.00	0.00
20																				0.00	0.00	0.00
21																					0.00	0.00
22																						0.00

From these results, we select candidates 2,8 and 13 as best candidates (1 is the use case itself). Bigger systems cannot be printed in this article, but the computations are cheap enough to allow for huge systems, with hundreds of candidates (notice that matrix $\hat{M}$ grows by power 2 in the number of candidate frameworks: $\mathrm{O}(M) = Q^2$.

6 Concluding remarks

In this paper, we have explored a self-consistent textual energy-based similarity measure for the comparison and ranking of frameworks. We first formalized a way to describe properties of frameworks. We used part-of-speech analysis to derive framework properties such as goal, propositional knowledge used, resources, application scope etc., form their meta data found on the internet. Earlier work has shown that more than 50% of frameworks can be classified this way. Quantification techniques have been developed to study distribution of framework properties and Markov chain models have been used to establish the likelihood of finding a framework with predefined properties. Frameworks not only differ in goals, resource, underpinning knowledge or application scope, they also differ in the context of the information and company architecture. Some focus on the managerial level, some on the shop floor level, some target the system level, some the technology level. We argued that the Zachman Framework can be used to frame frameworks as studied. When discussing product life cycles, additional models may have to be merged. We ultimately want to compile and study federations of frameworks and most likely, but not exclusively, such federations live on a single Zachman level or within a single Zachman viewpoint column.

We showed that by considering similarity in part of framework properties while varying a remaining part, one can mine for candidate frameworks for more or less compatible goals, application scopes, resources etc. This way, frameworks can be compared with and set off from alternative candidates, paired with related frameworks that integrate other Zachman aspects and finally we can integrate use case data. Framework designers can predefine framework modification limits in a contract, like with models. Contract-based framework transformations are modifications based on this contract, to extend existing frameworks properties and utility within the contract-specified boundaries. The similarity measure matches shared target terms among candidate frameworks and use cases. Framework descriptions carrying a common target term, or having a direct environment that connects them through a shared target term, develop a match with the paired framework. The higher the matching coefficient, the stronger the bond. The similarity measure explored in this paper has the capability to indicate similarity, its strength, its absence and the contribution of each of the considered candidates. When not matching at all, frameworks are called orthogonal. The similarity measure can also detect orthogonality. Unfortunately, the similarity measure studied can not measure dissimilarity in general. An absence of similarity is an indicator, but is generally incapable and insufficient to develop a quantified dissimilarity. As of yet, dissimilarity must be specified by framework designers in the framework transformation contract. In the future, it might also be derived from framework properties mined on the internet. Both contract-based framework transformations and the necessary dissimilarity measure(s) are subject of future research. To raise the effectiveness of framework mining a formal framework ontology would be helpful. This too is a subject for further research.

7 Bibliographie

[1] ANAND, G. and KODALI, R., *Development of a framework for lean manufacturing systems*. International Journal of Services and Operations Management, 5, 2009, pp. 687-716.

[2] BERENDS, J. & VAN TOOREN, M. , *Design of a Multi-Agent Task Environment Framework to support Multidisciplinary Design and Optimisation*. In , 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 2007.

[3] BERENDS, J., VAN TOOREN, M. and SCHUT, E., *Design and implementation of a new generation multi-agent task environment framework*. In , Proceedings of the 49th AIAA/ASME/ASCE/AHS/ ASC Structures, Structural Dynamics, and Materials, April 7-10, 2008, Schaumburg, IL, USA (p. 15).

[4] CHEN, C. , *Searching for intellectual turning points: Progressive knowledge domain visualization*. PNAS, 101, 2004, pp. 5305-5310.

[5] CHEN, C., *CiteSpaceII: Detecting and visualizing emerging trends and transient patterns in scientific literature.* Journal of the American Society for Information Science and Technology, 57, 2006, pp. 359-377.

[6] CHEN, C., IBEKWE-SANJUAN, F. and HOU, J. , *The Structure and Dynamics of Co-Citation Clusters: A Multiple-Perspective Co-Citation Analysis*. Journal of the American Society for Information Science and Technology, abs/1002.1985, 2010, p. 33.

[7] FERNANDEZ, S., SANJUAN, E. and TORRES-MORENO, J., *Textual Energy of Associative Memories: performants applications of Enertex algorithm in text summarization and topic segmentation*. In , Lecture Notes in Computer Science: MICAI'07, Aguascalientes, Mexico , 2007, p. 11.

[8] GAL, A., LAPALME, G., SAINT-DIZIER, P. and SOMERS, H. , *Prolog for Natural Language Processing*. 1991, John Wiley & Sons, Chichester, England.

[9] Gerritsen, B., *How to Adapt Information Technology Innovations to Industrial Design and Manufacturing to Benefit Maximally from Them*. Strojniski vestnik – Journal of Mechanical Engineering, 54, 2008a, pp. 426-445.

[10] GERRITSEN, B. , *IT innovations and their impact on industrial design and manufacturing*. In I. Horváth & Z. Rusák (Eds.), Proceedings of the Seventh International Symposium on Tools and Methods of Competitive Engineering - TMCE 2008 (pp. 63-77), 2008b, Delft University of Technology

[11] GERRITSEN, B., Engineering Frameworks: *A Bibliographic Survey-Based Problem Inventory*, Proceedings of the 1st IMS Summer School 2010, May 26-28, Zurich, Switzerland

[12] HECKEL, R. and LOHMANN, M. , *Towards Contract-based Testing of Web Services*. Electronic Notes in Theoretical Computer Science, 116, 2005, pp. 145-156.

[13] KODIYALAM, S. and SOBIESZCZANSKI-SOBIESKI, J., *Multidisciplinary design optimization - some formal methods, framework requirements, and application to vehicle design*. Int. J. Vehicle Design, 25, 2001, pp. 3-22.

[14] LAMAR, C. and MOCKO, G. *, Linguistic analysis of natural language engineering requirement statements* . In I. Horvath, F. Mandorli & Z. Rusak (Eds.), Proceedings of the TMCE 2010 Symposium (pp. 97-111)., 2010, Delft University

[15] MANNING, C. and SCHUTZE, H. , *Foundations of Statistical Natural Language Processing*. 1999, MIT Press.

[16] MOKYR, J. , *The Gifts of Athena; Historical origins of the knowledge economy*. 2005, Princeton University Press. Princeton, NJ, USA

[17] PADULA, S. and GILLIAN, R. *, Multidisciplinary Environments: A history of engineering framework development*. In , Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference - AIAA2006 (p. 11). 2006, NASA

[18] SALAS, A. and TOWNSEND, J. , *Framework requirements for MDO application development*. In , Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1998, St. Louis, MO, USA.

[19] SINCLAIR, J. , *Corpus, Concordance, and Collation*. 1991, Oxford Press, Oxford, UK

[20] STUMME, G. , *Off to new shores: conceptual knowledge discovery and processing*. Int. J. Human-Computer Studies, 59, 2003, pp. 287-325.

[21] SYNNESTVEDT, M., CHEN, C and HOLMES, J. , *CiteSpaceII: Visualization and Knowledge Discovery in Bibliographic Databases*. In , Proceedings of the 2005 Annual American Medical Informatics Association AMIA 2005 Symposium, Nov 30 - Dec 3, 2005, Austin, TX, USA (pp. 724-728).

[22] TARPEY, Th and FLURY, B., *Self-Consistency: A Fundamental Concept in Statistics*, Statistical Science, Vol. 11, 1996, pp. 229-243

[23] VAN GORP, P. and JANSSENS, D. , *CAViT: a consistency maintenance framework based on transformation contracts*. In , Dagstuhl Seminar Proceedings 05161, 2006.

[24] ZACHMAN, J. , *A Framework for Information Systems Architecture*. IBM Systems Journal, 26, 1987, pp. 276-292.

[25] ZANG, T., SAMAREH, J. and BAO, H., *Multidisciplinary Design Optimization Methods for Performance and/or Cost Optimization of Vehicles*. 2000