Identifying the Aspectual Requirements: A Theme Based Vector-Orientation Model

Dipankar Majumdar
Department of Information Technology
B.P. Poddar Institute of Management & Technology
137, V.I.P. Road. Kolkata – 700052, India

Swapan Bhattacharya
National Institute of Technology
M. G. Avenue, Durgapur
West Bengal – 713209, India

ABSTRACT

'Separation of Concerns' in the field of Software Engineering has been an important issue for quite some time. And this issue is very much related to Aspect Oriented Software Development. This is so because Aspects happen to be certain concerns that get interleaved with the Core-Functionalities in such a way that they become nearly inseparable. As a result of which both the designer as well as the programmer, who are supposed to be concerned only with the Core-Functionalities, is bound to take extra burden or botheration regarding the proper and accurate handling of Aspects. The Theme approach is an already established approach for Aspect identification in the requirements-engineering phase. Our approach is a diversification of the Theme approach where we look for Aspectual Requirements instead of Aspectual Themes. This paper proposes a purely mathematical model for Requirements-Engineering for Aspect Identification. The concept is based on N-Dimensional-Vector-Orientation Model, which is used to serve the purpose.

Categories and Subject Descriptors

D.2.1 **Requirement /Specifications**

General Terms

Theory, Measurement

Keywords

Requirements Engineering, Aspect-Oriented-Programming, Vector Orientation, N-Dimensional Space.

1. INTRODUCTION

Aspect Oriented Programming (AOP) has been an active research area recently for some years. In due course, the Aspects are identified as Concerns that gets tightly coupled with other Core-Functionality Modules and thereby to the system as a whole, making the system more and more complex to handle. These interfering requirements (Aspectual-Requirements) remain interleaved with the Core-Functionality Requirements in order to perform certain essential activities. Since these Aspectual Requirements does not form a logical part of these Core Functionality Requirements, a major objective of the 'Aspect Orientation' is to somehow separate the Aspectual Requirements from the Core Functionality Requirements.

On the other hand we cannot afford to ignore the functional necessity of these Aspectual Requirements. The objective is only to free the designer and the coder off the burden of handling the Aspectual Requirements, while designing the Core Functionality Requirements and not to totally ignore the Aspects as a whole. If the Aspectual-Requirements could be separated right at the Requirements Engineering phase, then the Aspectual Modules may be designed and coded separately under the Aspect Oriented Paradigm.

2. RELATED WORK

The objective of our work is to support requirements engineering for identification of aspects, and traceability of those aspects to (and from) the design from a mathematical point of view. Hence, our related work primarily describes work on identification of aspects from requirements. Previous publications on Theme/UML [4, 6, 5] describe other work on design. There have been several efforts in capturing and relating aspect-oriented requirements [16, 18, 8, 13, 11, 10, 3]. We shall consider the two, which relate most closely to the Theme approach. Rashid et al [13] provide the AORE (Aspect-Oriented Requirements Engineering) model and ARCaDe (Aspectual Requirements Composition and Decision support) approach and tool for describing components and requirements-level aspects. Examples of these aspects are compatibility, availability, or security. This work grows on the ViewPoints model [9], which is planned to support the integration of heterogeneous requirements specified from multiple perspectives. An early stage in the AORE model is the identification and specification of concerns. The approach to this differs from the Theme approach to concern identification in that it relies on the domain knowledge of the developer to identify possible non-functional requirements to be taken into account when implementing a particular requirement. Those concerns are not explicitly mentioned in the requirements specification; it is up to the developer to ascertain their relevance on their own. The Theme/Doc approach aims to support the analysis of relationships between behaviors described in requirements specifications. It is possible that the Theme/Doc approach to aspect identification could be used during the concern identification phase of AORE, or could support AORE's extension to include functional as well as non-functional requirements. Katera and Katz [11] propose architectural views of aspects as a means for reasoning about the relationships among aspects in a system. They describe aspects as crosscutting augmentations to an existing design. In particular, they allow for specification of the overlap between aspects through the concept of a *sub-aspect* that provides the overlapping functionality, and they make relationships between aspects explicit. A UML approach has been given to support the views, which differs from the Theme/UML approach: it provides additional architectural support for aspect modeling to that provided by Theme/UML, and it uses aspect mappings

rather than multi-dimensional composition style semantics. Several publications have been made in the past, which endorse the fact that Aspect Oriented Programming proves to be highly effective in comparison with the traditional Object Oriented Programming with respect to the issues like *Implementations of Cross Cutting Concerns, Error Debugging, Maintenance, Modularity, Reusability, Readability, Compactness and Developmental Effort.* Early in 1998, Murphy, Walker, and Beniassad [18] made an analysis on the usefulness and usability of the AOP Technology and made necessary postulates regarding the enhancement of the same. Tarr, Peri et al. proposed the idea of N Degrees of Separation: Multi-Dimensional Separation of Concerns in [7]. Siobhán Clarke and Elisa Baniassad in their book [15] introduced the concept of Themes. They proposed the idea how to find themes and thereafter look for aspectual themes. Our work is based on [15], the only difference being that we are looking for aspectual requirements and not themes. And we intend to separate these aspectual requirements right at the requirements engineering level in order to achieve an early separation of concerns.

3. SCOPE OF THE WORK
The scope of our work is to give a purely mathematical shape to the requirements engineering and aspect identification process. Once this can be done, the aspect identification task becomes completely mechanized and can be automated. We intend to identify the aspects early in the requirement-engineering phase, such that thereafter the aspectual requirements can be separated from the core-requirements and be placed under the AOP paradigm for further modeling and thereafter the implementation. As themes [20] happen to be already an established concept in the field of Aspect-Identification, we intend to give the theme-based approach a mathematical shape using vector analysis. We have defined an N-Dimensional space where N is the number of themes. Under this scenario we model the themes as mutually orthogonal co-ordinate axes and the requirements as N-Dimensional Unit Vectors.

By property of a unit-vector [17], we know that it has a magnitude equal to one and a direction. Likewise a 'requirement' may be represented by a unit magnitude and a direction, depending on the number of entities and the number of themes it involves for its functionality. Hence a requirement may get involved with single or multiple entities and at the same time single or multiple themes. Our requirement-vector shall have a magnitude equal to one and the direction or alignment in the vector space determined by the number of entities it involves and the number of themes it involves. The more a requirement gets involved with a theme, depending on the number of entities participating for that theme, the more is its inclination towards the axis represented by that very theme and lesser is the angle made by that requirement-vector with that theme-axis.

Consequently the direction-cosines of a requirement-vector with the theme-axes represent the involvement it has with them. Thus we see that while the 'magnitude' of the requirement-vector remains unity, the number of themes involved and the number of entities involved together determines its 'direction'. These two values together results in the requirement-vector having its base at the origin and making 'N' same/different angles with the theme-axes. Our definition of an Aspectual-Requirement is a requirement that gets involved with more number of themes.

Since we have clearly shaped the requirements and themes with a mathematical background, we can now easily devise a technique to find out those vectors that get involved with more number of axes. Such a vector represents a requirement that is involved with most of the themes and may be considered as an aspectual requirement from our definition.

Our work goes one step ahead in defining an N-Dimensional space and names it as the *'Probabilistic Aspect-Zone'*. Any requirement-vector appearing in that zone can be probabilistically said to be having aspectual characteristics.

3.1 Theme Based Vector Orientation Model
In this section we introduce the Theme Based-Dimensional-Approach analogous to the Theme-Approach [15] for Aspect-Oriented Analysis and Design. We use an 'Expression-Evaluation-System (EES)' as a case study borrowed from [15] to illustrate our proposed approach in Section 4.0. This approach can be used to portray an abstraction for the requirements, and conditionally express how to identify crosscutting aspectual-requirements. Our approach is different from the theme approach based aspect identification procedure in a sense that [15, 20] shows how to identify aspects from themes. On the other hand our approach is towards identifying Aspects from the 'Requirements'. We look into the 'requirements' in order to identify the 'aspectual-requirements' and not the aspectual-themes. The following are the salient points of our assumption.

- Themes are independent of each other
- Themes may interact with each other through the requirements
- A Requirement may get involved with one or more themes.
- A Requirement may not remain restricted to any particular theme
- A Requirement may not remain scattered into all the themes
- A Requirement may depend on one or more themes, but a theme never depend on any requirement

The Theme Based Vector Orientation Approach involves identifying the potential themes [15], which are then assembled together to form an N-dimensional space to prepare a design for the whole system. Themes can be thought of as analogous to a shortlisted list of the main actions or verbs, at its primitive level, that happens to characterize a system visibly. In our approach these themes are taken as reference axes mutually orthogonal to each other thereby giving rise to a hypothetical N-Dimensional space, where N is equal to the number of themes. And the requirements are fitted into this N-Dimensional space as N-Dimensional vectors.

So, in a sense, our Theme Based Vector Orientation Approach converts the 'Theme – Requirement' relationship to a purely mathematical model, where vector [17] based dimensional analysis can straight away be applied for Requirement-Engineering for the identification of Aspects.

3.2 Mathematical Analysis of the Approach
Let us consider an N-Dimensional Space, guided by N-Coordinate axes. Any vector having its base at the origin will have N dimensions. These kind of vectors [17] are represented by a 1 X N matrix as shown below:

Let V =

	1	2	3		...	i	i+1	...	N
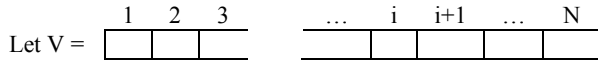
The above shows an N-Dimensional Vector represented by a 1 X N Matrix. The N-Dimensional vector V makes N angles with the N axes respectively. The magnitude of the vector and the angles it makes with the theme-axes are represented as follows:

$$|V| = \frac{V(1) \cdot U_1 + V(2) \cdot U_2 + \dots + V(i) \cdot U_i + \dots V(N) \cdot U_N}{\sqrt{[V(1)^2 + V(2)^2 + \dots + V(i)^2 + \dots V(N)^2]}}$$

$$\theta_{V\text{-}1} = Cos^{-1}\left(\frac{V(1)}{\sqrt{[V(1)^2 + V(2)^2 + \dots + V(i)^2 + \dots V(N)^2]}} \right)$$

$U_1, U_2, \dots U_N$ represents the unit-vectors along the theme-axes. And similarly $\theta_{V\text{-}2}, \theta_{V\text{-}3}, \dots$ upto $\theta_{V\text{-}N}$. Where $\theta_{V\text{-}i}$ is the angle that the vector: V makes with the ith coordinate axis. In the above expression we represent the 1st element of the vector V by V(1), 2nd element by V(2), 3rd element by V(3) and similarly upto V(N).

Further let us consider an omni-directional unit vector U, such that $\theta_{U\text{-}1} = \theta_{U\text{-}2} = \dots = \theta_{U\text{-}i} = \dots \theta_{U\text{-}N} \dots = \theta$ (say)
Any vector that gets involved with more number of the themes makes some positive angles, less than or equal to θ (depending on the number of entities involved while interacting with each theme), with each of these involving theme-axes.

The more number of themes and the number of entities the requirement-vector involves, the lesser is the angle between that requirement-vector (V for instance) and the unit-vector (U for instance). Thus we have the following.
Let φ be the angle between V and U, then

$$\varphi_{V\text{-}U} = Cos^{-1}[\theta_{V\text{-}1} \cdot \theta_{U\text{-}1} + \theta_{V\text{-}2} \cdot \theta_{U\text{-}2} + \dots + \theta_{V\text{-}i} \cdot \theta_{U\text{-}i} + \dots + \theta_{V\text{-}N} \cdot \theta_{U\text{-}N}]$$

Our hypothesis for an Aspectual-Requirement can be probabilistically laid down as follows:

"The lesser the angle a requirement-vector makes with the omni-directional unit vector, the higher is the probability of the vector being an Aspectual Requirement".

Mathematically we may represent the hypothesis as follows:

$p_{ASPECT}(R_i)$
= Probability of a Requirement being an Aspectual Requirement.

And $p_{ASPECT}(R_i) = \dfrac{(\theta - \varphi_{Ri\text{-}U})}{\theta}$

3.3. Probabilistic Aspect Zone
From the above, we draw the following corollary. Let us consider a vector **Z** anywhere in the N-Dimensional space that makes an angle equal to φ with the omni-directional unit vector U such that $\theta >= \varphi >= 0$. If the base of such a vector is kept static at the origin and the tip of this vector is rotated all through the N-Dimensions about the vector U keeping φ constant, we find an N Dimensional surface for a certain value of φ. Varying the angle φ from 0 to θ, we define this N-Dimensional space probabilistically.

The lesser the value of the angle φ, the greater is the probability of finding an aspectual-requirement there. We name this N-Dimensional space guided by the rotating vector **Z** as the ***'Probabilistic-Aspect-Zone'***.

We further propose that any requirement-vector appearing to be in this region is likely to be an aspectual requirement depending on the probabilistic value of the space. This requirement should be separately analyzed and studied and if required may be modeled as aspectual-requirement. And thereafter be treated separately in further requirement-engineering task.

4.0 Case Study
We now review the nature of themes [15], with the 'Expression Evaluation System (EES)' again borrowed from [15] as a case study for the demonstration of our approach. Table –1 shows the requirements of our Expression Evaluation System case study. Table-1 shows the entities in bold text and themes in bold-underlined text.

4.1 Finding Themes
The first step in the Theme Based-Vector Orientation Model is to examine the documentation of the system requirements. At this stage, we try to identify potential 'action-features' that are described in the requirements from the EES and also to find out which portions of the requirements document pertain specifically to those features. In Table-1, the identified potential themes are shown underlined.

Table-1: Expression Evaluation System Requirements

R1	An evaluation capability, which **determines** the result of **evaluating** an **expression**.
R2	A **display** capability, which depicts an **expression** textually
R3	A **check-syntax** capability, which optionally **determines** whether an **expression** is syntactically and semantically correct
R4	The **check-syntax**, **display**, and **evaluation** operations on any **expression** should all be **logged**

Table-1: Expression Evaluation System Requirements Continued..

R5	The **expression** is <u>defined</u> as a **variable-expression** or a **number-expression** or a **plus-operation** or a **minus-operation** or a **unary-plus-op** or a **unary-minus-op**
R6	**plus-operation** is <u>defined</u> as an **expression** and a **plus-operator** and an **expression**
R7	**minus-operation** is <u>defined</u> as an **expression** and a **minus-operator** and an **expression**
R8	**unary-plus-op** is <u>defined</u> as a **plus-operator** and an **expression**
R9	**unary-minus-op** is <u>defined</u> as a **minus-operator** and an **expression**
R10	**variable-expression** is <u>defined</u> as a letter and an **expression**
R11	**number-expression** is <u>defined</u> as a number and an **expression**

In object orientation, classes, or entities, form the main unit of modularity. In the Theme Based Vector Orientation Model, basic action-features are as much a unit of modularity as entities. The first step is to identify a set of primary action-features from our requirements. So, rather than just sifting through our requirements and looking for key entities, we also look for key themes.

In the next step, we iterate over that set, deciding whether to add, delete, split up, or group themes. As in Object Oriented Style, where we use at least some of the entities to motivate classes, we use some of the actions/verbs to motivate themes.

There are several ways to arrive at a starting point of Theme Based Vector Orientation Model. Names of 'Services', or 'Use Cases' of the system can be chosen to become potential themes.

In the case study namely the Expression Evaluation System (EES) borrowed from [15], we have no Use-Case described and have not analyzed the requirements in terms of services. Instead, we scan the requirements for identifiable pieces of functionality. We identify 6 potential themes:

Table-2: List of Themes

1	EVALUATION
2	DISPLAY
3	DETERMINE
4	CHECK-SYNTAX
5	LOG
6	DEFINE

We also identify 9 entities:

Table-3: List of Entities

1	EXPRESSION
2	VARIABLE-EXPRESSION
3	NUMBER-EXPRESSION

4	PLUS-OPERATION

Table-3: List of Entities Continued

5	MINUS-OPERATION
6	PLUS-OPERATOR
7	MINUS-OPERATOR
8	UNARY-PLUS-OP
9	UNARY-MINUS-OP

As told before we will consider the Requirements to be 'vectors' in a hypothetical-space with number of dimensions equal to the number of themes. Each of the vectors having their magnitude equal to unity and the direction depicting the involvement of the requirement in terms of the number of participating entities with one or more themes. In our case study, we have 6-Dimensions, and 11 Requirements to fit in. The list of the vectors with their magnitude being equal to unity, the direction cosines are shown is as follows. The coefficients of the requirement-vectors along each of the dimensions (theme-axes) are taken up as the number of entities involved for that 'requirement' with that theme.

Therefore for any requirement, we count the number of entities involved *for a particular theme in case of a particular requirement* from Table-1 and put them up as coefficients along the respective theme-axes. For example, we observe that for the Requirement: R1, the only involved entity 'expression' is working with two different themes, namely 'determine' and 'evaluate'. Therefore for R1 we have the requirement-vector having coefficient 1 with both the 1st and the 3rd dimensional unit-vector. As a result for R1, the coefficient of both **I** as well as **K** are equal to 1. We have used the convention $\theta_{RP\text{-}Q}$ to denote the angle between the 'vector' represented by the Pth Requirement and the Qth Axis. As convention the vectors and directional unit vectors are marked in bold.

$$\mathbf{R1} = \frac{1\cdot\mathbf{I} + 0\cdot\mathbf{J} + 1\cdot\mathbf{K} + 0\cdot\mathbf{L} + 0\cdot\mathbf{M} + 0\cdot\mathbf{N}}{\sqrt{(1^2 + 0^2 + 1^2 + 0^2 + 0^2 + 0^2)}}$$

$\theta_{R1\text{-}1} = Cos^{-1}(1/\sqrt{2})$, $\theta_{R1\text{-}2} = Cos^{-1}(0/\sqrt{2})$, $\theta_{R1\text{-}3} = Cos^{-1}(1/\sqrt{2})$

$\theta_{R1\text{-}4} = Cos^{-1}(0/\sqrt{2})$, $\theta_{R1\text{-}5} = Cos^{-1}(0/\sqrt{2})$, $\theta_{R1\text{-}6} = Cos^{-1}(0/\sqrt{2})$

$\mathbf{R2} = \dfrac{0{\cdot}\mathbf{I} + 1{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 0{\cdot}\mathbf{N}}{\sqrt{(0^2 + 1^2 + 0^2 + 0^2 + 0^2 + 0^2)}}$

$\theta_{R2\text{-}1} = \text{Cos}^{-1}(0/1),\ \theta_{R2\text{-}2} = \text{Cos}^{-1}(1),\ \theta_{R2\text{-}3} = \text{Cos}^{-1}(0/1)$
$\theta_{R2\text{-}4} = \text{Cos}^{-1}(0/1),\ \theta_{R2\text{-}5} = \text{Cos}^{-1}(0/1),\ \theta_{R2\text{-}6} = \text{Cos}^{-1}(0/1)$

$\mathbf{R3} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 1{\cdot}\mathbf{K} + 1{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 0{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 1^2 + 1^2 + 0^2 + 0^2)}}$

$\theta_{R3\text{-}1}$
$= \text{Cos}^{-1}(0/\sqrt{2}),\ \theta_{R3\text{-}2} = \text{Cos}^{-1}(0/\sqrt{2}),\ \theta_{R3\text{-}3} = \text{Cos}^{-1}(1/\sqrt{2})$
$\theta_{R3\text{-}4}$
$= \text{Cos}^{-1}(1/\sqrt{2}),\ \theta_{R3\text{-}5} = \text{Cos}^{-1}(0/\sqrt{2}),\ \theta_{R3\text{-}6} = \text{Cos}^{-1}(0/\sqrt{2})$

$\mathbf{R4} = \dfrac{1{\cdot}\mathbf{I} + 1{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 1{\cdot}\mathbf{L} + 1{\cdot}\mathbf{M} + 0{\cdot}\mathbf{N}}{\sqrt{(1^2 + 1^2 + 0^2 + 1^2 + 1^2 + 0^2)}}$

$\theta_{R4\text{-}1} = \text{Cos}^{-1}(1/2),\ \theta_{R4\text{-}2} = \text{Cos}^{-1}(1/2),\ \theta_{R4\text{-}3} = \text{Cos}^{-1}(0/2)$
$\theta_{R4\text{-}4} = \text{Cos}^{-1}(1/2),\ \theta_{R4\text{-}5} = \text{Cos}^{-1}(1/2),\ \theta_{R4\text{-}6} = \text{Cos}^{-1}(0/2)$

$\mathbf{R5} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 7{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 7^2)}}$

$\theta_{R5\text{-}1} = \text{Cos}^{-1}(0/7),\ \theta_{R5\text{-}2} = \text{Cos}^{-1}(0/7),\ \theta_{R5\text{-}3} = \text{Cos}^{-1}(0/7)$
$\theta_{R5\text{-}4} = \text{Cos}^{-1}(0/7),\ \theta_{R5\text{-}5} = \text{Cos}^{-1}(0/7),\ \theta_{R5\text{-}6} = \text{Cos}^{-1}(7/7)$

$\mathbf{R6} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 4{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 4^2)}}$

$\theta_{R6\text{-}1} = \text{Cos}^{-1}(0/4),\ \theta_{R6\text{-}2} = \text{Cos}^{-1}(0/4),\ \theta_{R6\text{-}3} = \text{Cos}^{-1}(0/4)$
$\theta_{R6\text{-}4} = \text{Cos}^{-1}(0/4),\ \theta_{R6\text{-}5} = \text{Cos}^{-1}(0/4),\ \theta_{R6\text{-}6} = \text{Cos}^{-1}(4/4)$

$\mathbf{R7} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 4{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 4^2)}}$

$\theta_{R7\text{-}1} = \text{Cos}^{-1}(0/4),\ \theta_{R7\text{-}2} = \text{Cos}^{-1}(0/4),\ \theta_{R7\text{-}3} = \text{Cos}^{-1}(0/4)$
$\theta_{R7\text{-}4} = \text{Cos}^{-1}(0/4),\ \theta_{R7\text{-}5} = \text{Cos}^{-1}(0/4),\ \theta_{R7\text{-}6} = \text{Cos}^{-1}(4/4)$

$\mathbf{R8} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 3{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 3^2)}}$

$\theta_{R8\text{-}1} = \text{Cos}^{-1}(0/3),\ \theta_{R8\text{-}2} = \text{Cos}^{-1}(0/3),\ \theta_{R8\text{-}3} = \text{Cos}^{-1}(0/3)$
$\theta_{R8\text{-}4} = \text{Cos}^{-1}(0/3),\ \theta_{R8\text{-}5} = \text{Cos}^{-1}(0/3),\ \theta_{R8\text{-}6} = \text{Cos}^{-1}(3/3)$

$\mathbf{R9} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 3{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 3^2)}}$

$\theta_{R9\text{-}1} = \text{Cos}^{-1}(0/3),\ \theta_{R9\text{-}2} = \text{Cos}^{-1}(0/3),\ \theta_{R9\text{-}3} = \text{Cos}^{-1}(0/3)$
$\theta_{R9\text{-}4} = \text{Cos}^{-1}(0/3),\ \theta_{R9\text{-}5} = \text{Cos}^{-1}(0/3),\ \theta_{R9\text{-}6} = \text{Cos}^{-1}(3/3)$

$\mathbf{R10} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 2{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 2^2)}}$

$\theta_{R10\text{-}1} = \text{Cos}^{-1}(0/2),\ \theta_{R10\text{-}2} = \text{Cos}^{-1}(0/2),\ \theta_{R10\text{-}3} = \text{Cos}^{-1}(0/2)$
$\theta_{R10\text{-}4} = \text{Cos}^{-1}(0/2),\ \theta_{R10\text{-}5} = \text{Cos}^{-1}(0/2),\ \theta_{R10\text{-}6} = \text{Cos}^{-1}(2/2)$

$\mathbf{R11} = \dfrac{0{\cdot}\mathbf{I} + 0{\cdot}\mathbf{J} + 0{\cdot}\mathbf{K} + 0{\cdot}\mathbf{L} + 0{\cdot}\mathbf{M} + 2{\cdot}\mathbf{N}}{\sqrt{(0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 2^2)}}$

$\theta_{R11\text{-}1} = \text{Cos}^{-1}(0/2),\ \theta_{R11\text{-}2} = \text{Cos}^{-1}(0/2),\ \theta_{R11\text{-}3} = \text{Cos}^{-1}(0/2)$
$\theta_{R11\text{-}4} = \text{Cos}^{-1}(0/2),\ \theta_{R11\text{-}5} = \text{Cos}^{-1}(0/2),\ \theta_{R11\text{-}6} = \text{Cos}^{-1}(2/2)$

4.2 Theme Based Aspect Identification

In this section, we demonstrate how our Theme Based Dimensional Approach may be used to identify Aspects, such that they could be separated right at the time of requirements engineering and treated separately. Let us define an omni-directional unit vector U.

We call it unit-vector because |U| = 1, and omni-directional because it makes equal angles with all the theme-axes. In our case, we have 6 dimensions; therefore our omni-directional unit vector makes equal angles with all the 6 axes.

Thus we have $\theta_{U\text{-}1}$, $\theta_{U\text{-}2}$, $\theta_{U\text{-}3}$, $\theta_{U\text{-}4}$, $\theta_{U\text{-}5}$, $\theta_{U\text{-}6}$ the angles that U makes with all the 6 axes equal to each other. This implies from the law of direction cosines of a vector [17] that if we assume that $\theta_{U\text{-}1} = \theta_{U\text{-}2} = \theta_{U\text{-}3} = \theta_{U\text{-}4} = \theta_{U\text{-}5} = \theta_{U\text{-}6} = \theta$ (say) then $6{*}\cos^2\theta = 1$. Therefore $\cos\theta = \sqrt{(1/6)}$ radians.

According to our proposed idea, the Requirements represented by vectors that are closer to the omni-directional-unit-vector U are assumed to be those requirements that tend to scatter into a number of themes. Consequently they are more probable to be aspectual requirements. Hence in order to identify the proximity that a requirement-vector has with the omni-directional unit vector, we calculate the angle between them.

For the ith requirement: Ri, we have the angle between Ri and U computed as follows.

$\varphi_{Ri\text{-}U}$
$= \text{Cos}^{-1}[\text{Cos}(\theta_{RX\text{-}1}){*}\sqrt{(1/6)} + \text{Cos}(\theta_{RX\text{-}2}){*}\sqrt{(1/6)}$
$\qquad + \text{Cos}(\theta_{RX\text{-}3}){*}\sqrt{(1/6)} + \text{Cos}(\theta_{RX\text{-}4}){*}\sqrt{(1/6)}$
$\qquad + \text{Cos}(\theta_{RX\text{-}5}){*}\sqrt{(1/6)} + \text{Cos}(\theta_{RX\text{-}6}){*}\sqrt{(1/6)}]$

$\Rightarrow \varphi_{Ri\text{-}U}$
$= \text{Cos}^{-1}[\{\sqrt{(1/6)}\} {*} \{\text{Cos}(\theta_{RX\text{-}1}) + \text{Cos}(\theta_{RX\text{-}2})$
$\qquad + \text{Cos}(\theta_{RX\text{-}3}) + \text{Cos}(\theta_{RX\text{-}4})$
$\qquad + \text{Cos}(\theta_{RX\text{-}5}) + \text{Cos}(\theta_{RX\text{-}6})\}]$

From the corollary shown in section 3.3, let us identify the '***Probabilistic Aspect Zone***' for our particular case study.

We have

$\varphi_{R1\text{-}U} = 0.9553 \Rightarrow p_{ASPECT}(\mathbf{R1}) = 0.1695$

$\varphi_{R2\text{-}U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R2}) = 0$

$\varphi_{R3\text{-}U} = 0.9553 \Rightarrow p_{ASPECT}(\mathbf{R3}) = 0.1695$

$\varphi_{R4\text{-}U} = 0.6155 \Rightarrow p_{ASPECT}(\mathbf{R4}) = 0.4649$

$\varphi_{R5\text{-}U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R5}) = 0$

$\varphi_{R6\text{-}U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R6}) = 0$

$\varphi_{R7\text{-}U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R7}) = 0$
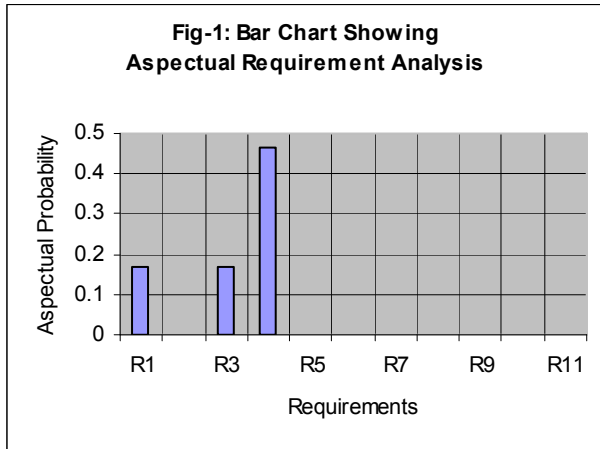
$\varphi_{R8-U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R8}) = 0$

$\varphi_{R9-U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R9}) = 0$

$\varphi_{R10-U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R10}) = 0$

$\varphi_{R11-U} = 1.1503 \Rightarrow p_{ASPECT}(\mathbf{R11}) = 0$

Fig-1: Bar Chart Showing Aspectual Requirement Analysis

From the bar-chart shown in Fig-1, we find that the requirement R4 has the highest probability of being an Aspectual Requirement. Such a requirement is required to be treated separately and processed under a separate programming paradigm, namely the Aspect Oriented Programming Paradigm. Likewise, we also note that the N-Dimensional space engulfing only the requirement R4 has higher Aspectual Probability than the space that engulfs R1 and R3 as well. We also find that since the rest of the requirements have their Aspectual-Probability

equal to 0, they are unlikely to be an Aspect and should be separately handled as non-aspectual-requirement.

Hence we put-forward the requirement as a probable aspectual requirement and recommend further study on it in order to conform that it truly satisfies all the condition of being an Aspect.

5. DISCUSSION AND CONCLUSION

Identification of Aspects is always necessary as early as possible in the phases of the Software Development Life Cycle. And therefore our objective was to identify the Aspectual Requirements as early as in the phase of Requirements Engineering. On examination of any conventional 'requirement', we found that they consist of a set of one or more entities performing certain actions.

From our acquaintance of 'themes', we have framed these actions as themes in our model. Also from our prior knowledge of Aspect Oriented Systems, we know that Aspects happen to be certain requirements that remain scattered irrespective of the domain of Entities or Themes. As a result, we look for those requirements that remain scattered throughout the domain of an information system.

In the current paper we have adopted and demonstrated a Theme-Based Vector Orientation Model for identification of Aspectual-Requirements and have proved its efficacy with a case study. Our approach is not only limited to theoretical studies but is practicable in case of large applications as well. Themes are an established concept in the field of Aspect Identification, but till date we do not have any core mathematical procedure for such identification task. In that respect our approach is expected to contribute in that area. And at the same time it is expected to give a core mathematical shape to the Aspect Orientation of Requirement Engineering.

6. REFERENCES

[1] Aspectj home page, Xerox PARC, USA, http://aspectj.org/.

[2] E. Baniassad, G. Murphy, and C. Schwanninger. Design pattern rationale graphs: Linking design to source. In *Proceedings of the International Conference on Software Engineering*, pages 352–362, 2003.

[3] J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering: The tropos project, 2002.

[4] S. Clarke. Extending standard uml with model composition semantics. *Science of Computer Programming*, 4(1):71–100, July 2002.

[5] S. Clarke and R.Walker. Towards a standard design languagefor AOSD. In *Proceedings of the International Conference on Aspect-Oriented Software Development*, pages 113–119.ACM Press, 2002.

[6] S. Clarke and R. J. Walker. Composition patterns: An approachto designing reusable aspects. In *International Conference on Software Engineering*, pages 5–14, 2001.

[7] Tarr, Peri, Harold Ossher, William Harrison, and Stanley M. Sutton, Jr. N Degrees of Separation: Multi-Dimensional Separation of Concerns. In the proceedings of the International Conference on Software Engineering (ICSE), 1999

[8] R. Darimont and A. van Lamsweerde. Formal refinement patterns for goal-driven requirements elaboration. In *Foundations of Software Engineering*, pages 179–190, 1996.

[9] A. Finkelstein. The viewpoints faq. *BCS/IEE Software Engineering Journal*, 11(1), 1996.

[10] J. Grundy. Aspect-oriented requirements engineering for component based software systems. In *4th IEEE International Symposium on Requirements Engineering*, pages 84–91.

[11] M. Katera and S. Katz. Architectural views of aspects. In *Proceedings of the International Conference on Aspect oriented Software Development*, pages 1–10, 2003.

[12] Kiczales, G, J.Lamping, A.Mendhekar, C.Maeda, C.Lopes, Jean-Marc Loingtier, and J.Irwin. Aspect-Oriented

Programming. In the proceedings of the European Conference on Object-Oriented Programming (ECOOP), 1997

[13] A. Rashid, A. Moreira, and J. Araujo. Modularisation and composition of aspectual requirements. In *Proceedings of the International Conference on Aspect-oriented Software Development*, pages 11–20, 2003.

[14]Murray R Spigel, "Vector Analysis and an Introduction to Tensor Analysis",Mc Graw Hill 1968-06-01, ISBN-10/ASIN: 07060228X

[15] S Clarke, E Baniassad, "Aspect-Oriented Analysis and Design: The Theme Approach", Publisher: Addison Wesley Professional, Pub Date: March 23, 2005, ISBN: 0-321-24674-8
[16] S. Sutton and I. Rouvellou. Modeling of software concerns in cosmos. In *Proceedings of the International Conference on Aspect-oriented Software Development*, pages 127–133, 2002.

[17]Erwin Kreyszig, "Advanced Engineering Mathematics", 9th-Edition,Ohio State University ISBN: 978-0-471-48885-9 ©2006

[18] X. Wang and Y. Lesperance. Agent-oriented requirements engineering using congolog and i*. In *Submitted to AOIS-2001, Bi-Conference Workshop at Agents 2001 and CAiSE'01.*, 2001.

[19] G. C. Murphy, R. J. Walker and E. L. Beniassad. Evaluating Emerging Software Development Technologies: Lessons Learned from Assessing Aspect Oriented Programming Technical Report TR-98-10, Dept of Comp. Sc, Univ. of British Columbia, 1998

[20] Elisa Baniassad, Siobhan Clarke, "Theme: An Approach for Aspect-Oriented Analysis and Design" in proceedings of International Conference of Software Engineering (ICSE), 2004