EXAMINING THE CONSTRUCTS AND COMPONENT
DIMENSIONS OF THE OPENACCESS PROJECT

by

Yijun Xu

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Engineering in Telecommunications Technology Management

Department of Systems and Computer Engineering,
Carleton University
Ottawa, Canada, K1S 5B6
August 2007

©Copyright 2007 Yijun Xu
NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.
ABSTRACT

The OpenAccess Coalition develops the OpenAccess standard and its reference database implementation for the purpose of providing interoperability among electronic design automation tools. This research answers three questions: (i) what multidimensional constructs can be identified by examining the development of the OpenAccess standard, (ii) What are the differences between the characteristics of the OA reference database implementation and the characteristics common to open source assets, (iii) what is the relationship between the constructs identified and the literature.

A multi-level single case study method was used to examine information about the development of the standard over seven years, answer the three research questions, and generate insights. Five multilevel constructs and their constituent dimensions were identified and linked to the existing literature. The results suggest that using open development processes for the reference database implementation of the OpenAccess standard resulted in the production of assets that do not comply with the definition of open source. Top management teams may benefit from this research by better understanding the characteristics of the projects that develop open standards. This may help them interact with this type of projects more effectively. Academics may use the insights generated from this research to define and test hypotheses.
ACKNOWLEDGEMENTS

I greatly acknowledge my thesis supervisor Professor Stoyan Tanev for his tireless guidance, encouragement and support throughout this study. I have also greatly appreciated Professor Tony Bailetti’s help and invaluable support throughout the time of my Graduate studies and in the final stages of this study. I express my appreciation to Professor John Callahan and the rest of the Faculty team of the Technology Innovation Management (TIM) program at Carleton University for their generous comments and constructive suggestions at various stages of my work.

I thank my parents, Liying Shan and Senmu Xu, and my husband, Wen Lin, for their unconditional love and continuous encouragement throughout this study, for supporting me to pursue my interests and believing in me when I doubted myself.

The completion of this research would have been impossible without the support and close cooperation of the management team of the Silicon Integration Initiative (Si2), especially: Steve Schulz, President and CEO, Sumit DasGupta, Sr. Vice President of Engineering, and Donald R. Cottrell, Fellow: Emerging Technologies.

This thesis was developed as part of the Competing in Open Environments research program led by Tony Bailetti, Carleton University, and Peter Carbone, Nortel. The support received from Nortel for this research and the Technology Innovation Management program is gratefully acknowledged.
# TABLE OF CONTENTS

ABSTRACT .........................................................................................................................III

ACKNOWLEDGEMENTS ............................................................................................IV

1 INTRODUCTION ........................................................................................................1
  1.1 Objective.................................................................................................................2
  1.2 Deliverables ...........................................................................................................2
  1.3 Contributions .........................................................................................................3
  1.4 Relevance ................................................................................................................3
  1.5 Organization ..........................................................................................................4

2 LITERATURE REVIEW ..........................................................................................5
  2.1 Case study and longitudinal field research methodologies .....................................5
    2.1.1 Single case studies ....................................................................................5
    2.1.2 Multi-level longitudinal research on organizational change .......................7
  2.2 Research on open source projects ........................................................................9
    2.2.1 The definition of open source .......................................................................9
    2.2.2 User innovation network characteristics of open source .........................10
    2.2.3 Open source projects’ success measures ..............................................13
    2.2.4 Motives for contributing to open source projects ....................................15
    2.2.5 Type of market offers enabled by open source software .........................17
  2.3 Open standards .....................................................................................................18
  2.4 Diffusion of innovations and technology adoption ...........................................21
    2.4.1 Diffusion of innovation .............................................................................21
    2.4.2 Technology adoption life cycle ...................................................................23
  2.5 Lessons Learned ..................................................................................................24

3 RESEARCH METHOD ...........................................................................................27
  3.1 Unit of analysis ....................................................................................................27
  3.2 Study period ..........................................................................................................27
  3.3 Research method ..................................................................................................27
  3.4 Research steps ......................................................................................................28
  3.5 Collecting data and preparing a case study .......................................................29
  3.6 Identify research questions ..................................................................................31
  3.7 Identification of four layers ................................................................................31
  3.8 Identification of variables ...................................................................................33
  3.9 Identification of linkages between events and variables ..................................36
  3.10 Identification of constructs ..............................................................................37
  3.11 Differences between the OA RDBI and open source assets .............................37
  3.12 Relate the answers to the research questions to existing literature ................37

4 RESULTS .....................................................................................................................38
  4.1 A short description of the OA project based on the case study .............................38
    4.1.1 Project objective.......................................................................................38
    4.1.2 Three building blocks of the OA project ..............................................39
    4.1.3 OA governance structure .....................................................................40
    4.1.4 RDBI development process ..................................................................41

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
4.1.5 OA license terms .................................................................................... 42
4.2 Chronological timeline of events in the Si2/OAC decisions layer .......... 43
4.3 Chronological timeline of events in the OA members' activity layer ....... 44
  4.3.1 Si2/OAC membership ............................................................................ 46
  4.3.2 Number of OA EDA design tools ........................................................ 48
  4.3.3 Cadence's motives for and risks from releasing its technology ....... 49
  4.3.4 Motives and risks for companies adopting the OA standard ............ 50
4.4 Chronological timeline of events in OA RDBI layer ......................... 51
4.5 Chronological timeline of events in OA API layer ............................... 53
4.6 Cross-layer analysis .................................................................................. 56
  4.6.1 Linkages between events across the four layers ................................. 56
  4.6.2 Linkages between variables across the four layers ............................ 60
5 DISCUSSION OF RESULTS ........................................................................ 63
  5.1 Answers to research questions ................................................................. 63
    5.1.1 Multidimensional constructs of the OpenAccess standard ............... 63
    5.1.2 Differences between the OA RDBI and open source assets ............ 77
    5.1.3 Multidimensional constructs and research literature ....................... 80
  5.2 Insights gained while carrying out the research ...................................... 85
6 CONCLUSIONS ............................................................................................. 89
  6.1 Conclusions ............................................................................................. 89
  6.2 Limitations ............................................................................................. 90
  6.3 Future research ....................................................................................... 90
REFERENCES .................................................................................................... 91
APPENDICES .................................................................................................... 96
LIST OF TABLES

Table 2-1: Open source project success measures .............................................. 13
Table 2-2: Motives for companies to release code as open source .................. 15
Table 2-3: Motives of companies to participate in OSS development .............. 16
Table 2-4: Types of market offers in the open source software literature ....... 17
Table 2-5: Open standards definition ................................................................. 20
Table 2-6: Characteristics of a technology influencing its diffusion rate .......... 22
Table 3-1: A summary of Si2 data sources ....................................................... 30
Table 3-2: Variables in the OA members’ activity layer .................................... 34
Table 3-3: Motives and risks to participate in the OA project ......................... 34
Table 3-4: Company categories ........................................................................ 35
Table 3-5: Variables in the OA RDBI layer ...................................................... 35
Table 3-6: Variables in the OA API layer ......................................................... 36
Table 4-1: OA license terms ............................................................................... 42
Table 4-2: Description of the events in the Si2/OAC decision layer ............... 43
Table 4-3: Description of the events in the OA members’ activity layer .......... 45
Table 4-4: Summary of Si2/OAC membership & OA-based design tools ... 49
Table 4-5: OAC members’ motives to adopt the OA standard ......................... 51
Table 4-6: Description of the events in the OA RDBI layer ......................... 52
Table 4-7: Variables associated the OA RDBI layer ........................................ 53
Table 4-8: Description of the events in the OA API layer ............................. 54
Table 4-9: Descriptive statistics in OA API specification layer ..................... 55
Table 5-1: Constructs drawn from the Open Access project ......................... 64
Table 5-2: Number of contribution records by year .................................... 73
Table 5-3: OSI OS definition vs OA RDBI license terms ................................. 78
LIST OF FIGURES

Figure 1.1: Three building blocks of OA project ................................................. 1
Figure 4-1: OpenAccess project structure .......................................................... 40
Figure 4-2: Chronological timeline of events in the Si2/OAC decision layer . 43
Figure 4-3: Events in the OA members’ activity layer ..................................... 44
Figure 4-4: Si2 and OAC membership by year .................................................. 47
Figure 4-5: Newly announced OA EDA tools per year .................................... 48
Figure 4-6: Chronological timeline of events in the OA RDBI layer ............... 52
Figure 4-7: Chronological timeline of events in the OA API layer ................. 54
Figure 4-8: Chronological timeline of all events in the 4 layers ................. 56
Figure 4-9: Chronological timeline of simultaneous events ... 57
Figure 4-10: Chronological timeline of related events ................................. 60
Figure 4-11: Relationship between API & RDBI releases .......... 60
Figure 4-12: Accumulated downloads, membership and OA tools by year ... 61
Figure 5-1: The pedigree of the OpenAccess standard ................................ 66
Figure 5-2: Rights to OA standard specs and RDBI by member status ...... 67
Figure 5-3: Three phases of the OA project’s life cycle .................................. 68
Figure 5-4: The OA project’s life cycle Phase I .............................................. 69
Figure 5-5: The OA project’s life cycle Phase II ............................................. 70
Figure 5-6: The OA project’s life cycle Phases III .......................................... 71
Figure 5-7: Number of OA standard downloads ......................................... 74
LIST OF APPENDICES

Appendix A: The OpenAccess project case study.................................................96
Appendix B: Top five EDA companies by market value 2005 (US$)..............117


1 INTRODUCTION

The lack of interoperability between electronic design automation (EDA) tools was a major limitation for the cost-effective design and manufacturing of silicon integrated circuits. The Silicon Integration Initiative (Si2) is an organization comprised of 113 companies in the semiconductor, electronic systems and electronic design automation (EDA) tool industries. In late 1999, the Si2 established the OpenAccess (OA) project to develop a standard for an application programming interface for EDA tools and gain the adoption of the standard within the EDA industry. The OA project was designed to deliver interoperability among EDA tools and not just data exchange.

Figure 1.1 illustrates the three building blocks of the OA project.

Figure 1.1: Three building blocks of OA project

Source: OpenAccess™ Release 2.2 Standard API Tutorial, 4th Edition (OA 2.2.6) (with permission of the Si2)

1 Source: 2006 OAC membership list.
The OA standard specification includes three components that define the API: (i) an information model, (ii) a data model defined by C++ header files, and (iii) the API specifications which provides the header information in a more readable format. The OA reference database implementation (RDBI) is the code that implements the OA standard specification. The OpenEvolution process is the process that governs the development, distribution, and support of the OA standard.

1.1 Objective

The objective of this research is to answer three research questions and produce insights. The questions are:

- What are the multidimensional constructs that could be used to describe the development of the OpenAccess standard?
- What are the differences between the characteristics of the OA reference database implementation and the characteristics common to open source assets?
- What is the relationship between the constructs used to describe the OpenAccess standard and the research literature?

1.2 Deliverables

The deliverables of this research are:

- a case study anchored around the development of the OA standard
- an analytical chronology of the OA project
• answers to the three research questions
• insights generated from examining the characteristics of the OA project

1.3 Contributions

This research makes at least two contributions to the emerging literature on open ecosystems.

First, the research identifies five multidimensional constructs and their constituent dimensions for projects that develop open standards.

Second, this research identifies the differences between using open development processes to produce (i) an open standard and (ii) open source code.

1.4 Relevance

This research is relevant for at least three reasons. First, researchers who study competition in open environments may be interested in the approach used in this study to make sense of what is occurring around various open source phenomena going beyond software code development.

Second, this research may help top management teams of technology and end user companies to improve their interactions with projects that develop open standards.
Third, this research may help managers assess whether or not, and when to join an open standard organization such as the OpenAccess Coalition.

1.5 Organization

This thesis is organized into six chapters. Chapter 1 is the introduction. Chapter 2 provides the literature review and identifies the lessons learned from preparing it. Chapter 3 describes the research method. Chapter 4 presents the results of the research. Chapter 5 discusses the results of the research. Chapter 6 provides the conclusions, limitations, and suggestions for future research.
2 LITERATURE REVIEW

This chapter is organized into six sections. Section 2.1 reviews the literature on longitudinal field research methodologies. Section 2.2 provides a review of open source projects. Section 2.3 examines the open standards literature. Section 2.4 examines the technology adoption and innovation diffusion literature. Finally, Section 2.5 identifies the lessons learned from the literature review.

2.1 Case study and longitudinal field research methodologies

2.1.1 Single case studies

A case study is an empirical enquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident and it relies on multiple sources of evidence (Yin, 1994, p. 13). This methodology is ideally suited for when a holistic, in-depth investigation is needed (Eisenhardt, 1989; Feagin, Orum, & Sjoberg, 1991).

Yin (1994) identifies that there can be single-case or multiple-case applications in exploratory, explanatory, and descriptive types of case studies. Selecting between multiple or single cases is a critically important choice (Yin, 1994; Eisenhardt, 1991). A single-case study may be used to confirm or challenge a theory, or to represent a unique or extreme case. It is also ideal for revelatory cases where an observer may have access to a phenomenon that was previously inaccessible (Yin, 1994).
In a recent study Siggelkow (2007) points out that if one uses a single case study, the research cannot just focus on description, but also has to provide a conceptual insight. According to Siggelkow, there are at least three important uses for single case studies: to motivate a research question, to inspire emerging theory, and to be employed as illustration.

A key strength of the case study method involves consideration of construct validity, internal validity, external validity, and reliability (Yin, 1994). Yin (1994) suggested using multiple sources of evidence as the way to ensure construct validity. Typically, case study based research uses multiple sources of evidence such as ethnographic observation, in-depth interviews (in particular, oral histories), and historical documents. The specification of the unit of analysis is important for the internal validity as the theories are developed and data collection and analysis are used to test those theories. Both qualitative and quantitative research methods may be used (Yin, 1994). External validity is more difficult to attain in a single-case study. Yin (1994) suggests that external validity could be achieved from theoretical relationships on which generalizations could be made.
2.1.2 Multi-level longitudinal research on organizational change

Theoretically sound and practically useful research on change should explore the contexts, content, and process of change together with their interconnections through time (Pettigrew, 1990). An approach that offers both multi-level (vertical) and processual (horizontal) analysis is contextualist in character (Pettigrew, Woodman, & Cameron, 2001). The vertical level refers to the interdependences between higher or lower levels of analysis of the phenomena. The horizontal level refers to the sequential interconnectedness among phenomena in historical, present, and future time (Pettigrew, 1990).

Although dynamism has been difficult to study, the cross-sectional analysis of sequential static states leads to the understanding of the dynamics of change across time and space (Pettigrew, 1990). Weick and Quinn (1999) explained the change process as movement from one state to another. To describe the process of change through the evolution of states of events, it is quite common that process is encapsulated in the form of variables. Attention must be paid however since variables are not a one-to-one representation of the process itself and processes are closer to the actual behavior than are the variables (MacKenzie, 2000).

Sandelands and Drazin (1989) highlighted the limitation of the variables paradigm and instead proposed that theorizing about process is enabled by an active, dynamic vocabulary. Van de Ven (1995) argued that, in the literature on change, the process
should refer to a sequence of events that describe how things change over time. At the most general level, questions such as what, who, where, when, why, and how are the special features of a process vocabulary that can help uncover novel theoretical questions and reveal original findings (Pettigrew et al., 2001). Temporal history needs to be understood as events, chronology, structures, and underlying logic (Pettigrew et al., 2001).

In summary, Pettigrew (1990) described three background assumptions on the multi-level, longitudinal contextual research including: 1) every change should be studied within the context of changes happening at other levels of analysis; 2) the contextualist nature of the research method is manifested through the importance of revealing temporal interconnectedness; and 3) a context can be analyzed as a self-consistent arrangement of structures and processes where i) the processes influence the subjective interpretations of the involved actors and ii) the subjective interpretations of actors’ perceiving, learning, and remembering, help the shaping of the processes.

The research output from a longitudinal single case-study work may be in one of the following forms: 1) a case as an analytical chronology which tells the story across different levels of analysis; 2) a diagnostic case which contains the features of an analytical chronology but, in addition, contains a listing and analysis of the organization’s current strategic concerns, and; 3) an interpretative/theoretical case.
which interprets the narrative and links emerging conceptual and theoretical ideas inductively derived from the case both to stronger analytical themes within the case and wider theoretical debates in the literature (Pettigrew, 1990).

2.2 Research on open source projects

2.2.1 The definition of open source

The term “open source” has two meanings. First, “open source” refers to a specific and relatively well-defined approach to the licensing of intellectual property in software programs. Second, the term “open source” is also used to denote a software development process and business strategy that are not well defined (Hope, 2004).

According to the Open Source Initiative (www.opensource.org), the distribution terms of open-source software must comply with the following criteria:

- free redistribution, i.e., the license should not restrict any party from royalty free selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources
- the program must include source code, and must allow distribution in source code as well as compiled form
- the license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software
• integrity of the author’s source code - the license must explicitly permit
distribution of software built from modified source code
• the license must not discriminate against any person or group of persons
• the license must not restrict anyone from making use of the program in a
  specific field of endeavor - business, genetic research etc.
• the rights attached to the program must apply to all to whom the program is
  redistributed without the need for execution of an additional license by those
  parties
• license must not be specific to a product
• license must not restrict other software that is distributed along with the
  licensed software
• license must be technology-neutral, i.e no provision of the license may be
  predicated on any individual technology or style of interface.

2.2.2 User innovation network characteristics of open source development

There has recently been a surge of interest in assessing the feasibility of extending the
open source principles beyond the software domain (Franke & Shah, 2003; von Hippel
& von Krogh, 2003; Shah, 2005; von Hippel, 2005; Hienerth, 2006; Shah, 2006;
Baldwin & Clark, 2006; Goldman & Gabriel, 2005; Chesbrough & Crowther, 2006).
The success of open source as a methodology for technology development has been
influential in fields as diverse as astronomy (Ferris, 2002), automobiles (Franz, 2005),
sports equipment (Franke & Shah, 2003; Shah, 2000), personal computers (Freiberger & Swaine, 2000), video games (Jeppesen & Molin, 2003), mountain biking (Luthje et al., 2005), music instruments (Jeppesen & Frederiksen, 2006), and biotechnology (Hope, 2004). Hope (2004) argued that “open source” can be both desirably and broadly adopted by any industry participant, from an individual developer through to a major multinational firm (Hope, 2004, p. 92-93). She also pointed out the need of a more general definition of the term “open source” applicable to technology fields other than software development. For example, Bailetti (2006) defined an open source technology as an asset (e.g., software code, hardware designs, content, data format), with a distribution license that provides users the freedom to use the asset for any purpose, to study and modify the asset, and to redistribute copies of either the original or the modified asset, without having to pay royalties to its previous developers.

From a user innovation perspective, open source software development is an example of a horizontal user innovation network and can be identified by the presence of five characteristics (von Hippel & von Krogh, 2001; von Hippel, 2002; Meyer, 2003; Benkler, 2002):

1. Free revealing
2. User innovation community
3. Collective invention
4. Common-based peer production
5. User community support
Free revealing

Free revealing refers to a unique non-proprietary strategy granting access under the
terms of a license to all interested agents without imposing any direct payment (von
Hippel, 2002; Dietmar, Henkel, & von Hippel, 2003; Henkel, 2004; Baldwin & Clark,
2006).

User innovation community

A user innovation community refers to a group of individuals or companies that use
and improve the freely revealed innovation expecting to derive benefits, either from
the innovation itself or from the process of innovating (von Hippel, 1988).

Collective invention

Collective invention refers to a cyclic process of follow-on innovation leading to a
series of freely revealed incremental improvements to the technology or asset
triggering new rounds of innovation activities. Meyer (2003) identified three main
conditions which favor the development of a collective invention regime in any
industry including: i) a technological change or other new opportunity that may result
in a shift in the locus of competition, leading to sharing in relation to the new
non-competitive knowledge; ii) the degree of technological standardization that is
sufficient enough to facilitate the exchange of information among potential innovators;
and iii) technological uncertainty regarding the newly adaptable invention.
**Commons-based peer production**

According to Benkler (2002), peer production of information is a mode of production in which the members of a user community collaborate to improve or complete new versions of the released technology or asset.

**User community support**

Although they often co-exist, user innovation networks and user communities are independent phenomena (Hope, 2004). User communities are not always innovative (Muniz & O’Guinn, 2001). As an important feature of the open source development model, their governance and management mechanisms establish and maintain an effective community structure that maximizes other users’ motivation to contribute to further development (Lerner & Tirole, 2002).

### 2.2.3 Open source projects’ success measures

Table 2-1 lists the success measures of open source projects identified in the literature.

<table>
<thead>
<tr>
<th>Measure of Success</th>
<th>Indicators</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development process</td>
<td>• The number of downloads</td>
<td>Rehman (2006)</td>
</tr>
<tr>
<td>Project output</td>
<td>• The number of developers</td>
<td>Crowston, Annabi &amp; Howison (2003)</td>
</tr>
<tr>
<td></td>
<td>• The number of contributions</td>
<td>Crowston et al. (2004)</td>
</tr>
<tr>
<td></td>
<td>• The number of releases</td>
<td>Crowston &amp; Scozzi (2002)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rehman (2006)</td>
</tr>
</tbody>
</table>
Number of downloads

According to the result of an on-line survey, the success of an open source project is related to the number of downloads of the open source software (Rehman, 2006).

Number of developers

Projects involving a larger number of developers seem to be more successful (Crowston & Scozzi, 2002; Healy & Schussman, 2003). With more developers working on the project, problems are quickly fixed and the rate of software development is increased (Raymond, 1999).

Number of contributions

The success of open source software projects seems to depend on the continued contribution of volunteer developers (Crowston et al., 2003; Crowston, Annabi, Howison, & Masango, 2004). Open source software success was measured by the level of activity of developers in submitting code and bug reports (Crowston et al., 2003, 2004).

Number of releases

From the perspective of software developers, the success of a project may be measured in terms of the number of times the software is released in a given period (Crowston & Scozzi, 2002; Rehman, 2006).
2.2.4 Motives for contributing to open source projects

Table 2-2: Motives for companies to release code as open source

<table>
<thead>
<tr>
<th>Motive</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Increase the demand for complementary products</td>
<td>Companies can benefit if they can offer products and services in proprietary segments of the market which complement the OSS</td>
<td>Lerner &amp; Tirole, 2002; Lerner &amp; Tirole, 2004; Goldman &amp; Gabriel, 2005; Henkel, 2004</td>
</tr>
<tr>
<td>(2) Set standard</td>
<td>The release of code as open source develops network externalities that drive the establishment of a standard.</td>
<td>Goldman &amp; Gabriel, 2005; Henkel, 2004; Koenig, 2004</td>
</tr>
<tr>
<td>(3) Weaken competitors</td>
<td>Releasing code as open source helps the company pursue new markets, position itself against established competitors</td>
<td>Lerner &amp; Tirole, 2002; Henkel, 2004; Koenig, 2004</td>
</tr>
<tr>
<td>(4) Build better relations with customers</td>
<td>By releasing code as open source, commercial companies may generate good public relations with customers by inviting them to fix bugs and contribute enhancement</td>
<td>Goldman &amp; Gabriel, 2005; Henkel, 2004</td>
</tr>
<tr>
<td>(5) Benefit from external support</td>
<td>Companies release code as open source to receive design and development support from other firms and external developers or users</td>
<td>Goldman &amp; Gabriel, 2005; Henkel, 2004</td>
</tr>
<tr>
<td>(6) Nurture innovation</td>
<td>By releasing code as open source to find out what potential customers do with the code, company can explore new products</td>
<td>Goldman &amp; Gabriel, 2005</td>
</tr>
<tr>
<td>(7) Gain reputation</td>
<td>Companies release high quality OSS expecting to improve the company's technical reputation</td>
<td>Henkel, 2004</td>
</tr>
<tr>
<td>(8) Trigger gift economy effect</td>
<td>Provide gifts that trigger gift economy effect which ultimately encourages others to work in for the company</td>
<td>Gabriel &amp; Goldman, 2005</td>
</tr>
</tbody>
</table>

Motives for companies to release code as open source

Table 2-2 summarizes the motives that drive companies to release their proprietary code as open source.
Table 2-3: Motives of companies to participate in OSS development

<table>
<thead>
<tr>
<th>Motives</th>
<th>Brief description</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Increase the demand for complementary products</td>
<td>Firms participate in open source activities to generate revenues from the sale of related products</td>
<td>Bonaccorsi &amp; Rossi, 2004; Henkel, 2004; Lerner &amp; Tirole, 2002; Lerner &amp; Tirole, 2004; von Hippel &amp; von Krogh, 2003</td>
</tr>
<tr>
<td>(2) Nurture innovation</td>
<td>Open source development enables user innovation networks where users develop, produce, distribute and consume software.</td>
<td>Bonaccorsi &amp; Rossi, 2004; Goldman &amp; Gabriel, 2005; Raymond, 1999; von Hippel, 2002; von Hippel &amp; von Krogh, 2003</td>
</tr>
<tr>
<td>(3) Use OSS as a low cost component</td>
<td>Firms participate in the production of commodity components of their product line as a method of reducing their development and maintenance costs</td>
<td>Bonaccorsi &amp; Rossi, 2004; Fink, 2003; Hawkins, 2004</td>
</tr>
<tr>
<td>(4) Set standard</td>
<td>Adopting open source models helps develop network externalities and imposes a standard in network industries. This is particularly important for infrastructure and tools.</td>
<td>Goldman &amp; Gabriel, 2005; Koenig, 2004</td>
</tr>
<tr>
<td>(5) Implement strategy</td>
<td>Firms release their code as OSS to weaken competitors or decrease dependencies on suppliers</td>
<td>Henkel, 2004; Koenig, 2004; Lerner &amp; Tirole, 2002;</td>
</tr>
<tr>
<td>(6) Benefit from external support</td>
<td>Feedback and contributions from the open source community allow firms to lower software development and R&amp;D costs and improve their software through the bug fixing activities.</td>
<td>Bonaccorsi &amp; Rossi, 2004; Goldman &amp; Gabriel, 2005</td>
</tr>
<tr>
<td>(7) Make money on complementary services</td>
<td>Making money on complementary services such as support services instead of selling software.</td>
<td>Bonaccorsi &amp; Rossi, 2004</td>
</tr>
<tr>
<td>(8) Gain reputation</td>
<td>In addition to building reputation as a good open source player, releasing high quality OSS will also improve the company’s technical reputation</td>
<td>Henkel, 2004</td>
</tr>
</tbody>
</table>

Motives for companies to participate in open source software development

Motivations for contributions to open source software projects constitute a core issue in open source software research (von Krogh & von Hippel, 2006). The motives of
individuals and firms contributing to open source development are different (Bonaccorsi & Rossi, 2004). Table 2-3 lists the motives for companies to participate in open source development projects.

2.2.5 Type of market offers enabled by open source software

A product or market offer is anything that is offered to the market for attention, acquisition, use, or consumption that might satisfy a want or need (Kotler & Turner, 1995).

Alam (2006) examined how open source software projects contribute to companies' market offers and change their competitive environments. He identified six types (see Table 2-4) of market offers: (i) hosting and content services, (ii) training and testing services, (iii) support services, (iii) subscription services, (iv) commercial licenses, and (v) products and applications.

<table>
<thead>
<tr>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hosting and content services</td>
<td>A content delivery network that allows content creators to outsource delivery of their digital multimedia content. Content creators pay the company to host, manage, and deliver the content to their intended audience. Content services deliver ad-supported for free and subscription to premium content to end-users through the consumer website maintained by the company</td>
</tr>
<tr>
<td>Training and testing services</td>
<td>Provide professional certifications for those who want to show they have the skills, knowledge, and capability to work with the software. The companies tests and certify hardware and software from their partners. When the partners certify their products against the company’s software, this gives assurance to customers that the products integrates and works effectively together</td>
</tr>
</tbody>
</table>
Support services
Companies provide support services to their customers of open source software. They provide fixes to the software, advice and suggestions for deploying the software in their customers’ environment. The companies also provide custom development to meet certain requirements from their customers.

Subscription services
Provide subscription services to their customers. Customers subscribe to receive updates, upgrades, monitoring and management capabilities of their software deployments. They continue to receive these services as long as they are subscribed. Companies do not only offer subscription to updates but also on complements, e.g., VRT Certified Rules, critical to the software.

Commercial licenses
For customers, that do not or cannot abide by the open source license, they can purchase commercial licenses from the companies that have undisputed rights to the software. When customers purchase commercial licenses, the company may also allow certain proprietary assets to be integrated into the customers’ applications. Customers may also use trademarks owned by the company.

Products and applications
Companies develop products and applications that integrate or complement the open source software to create a system. The products and applications may provide advanced features and capabilities over what is available in the open source version. The companies then sell these products bundled with other add-ons to their customers.

Source: Alam (2006)

2.3 Open standards

The Si2 defines an open standard as: “fully documented specifications or software code, managed by a community of stakeholders using a democratic group process that has been granted derivative works rights, that is available to all interested parties under non-discriminatory terms, and which can be observed, used and designed to with full fidelity” (Open DFM Workshop, 2006).

The emerging open standards literature does not fully agree on what the term “open standard” actually means. An open standard is more than just a specification (Tiemann, 2006). The principles behind the standard, and the practice of offering and operating the standard, are what make the standard open (Perens,
Researchers (von Burg, 2001, p. 34; West, 2003, 2004, 2005; Krechmer, 2005; Tiemann, 2006) have identified multiple intermediate levels of openness by means of a number of examples including predominantly proprietary standards and predominantly open standards.

Based on a software development perspective, Perens (http://perens.com) identifies six principles associated with open standards: availability, maximize end-user choice, no royalty, no discrimination, extension or subset and predatory practices. Combining very different views of the creators, implementers and users of standards on what is an open standard Krechmer (2005) suggests ten requirements that enable the existence of open standards: open meeting, consensus, due process, open IPR, one world, open change, open documents, open interface, open access, on-going support. A reference implementation of a standard can be extremely valuable in identifying gaps or hidden assumptions that may underlie the development of that particular standard (Tiemann, 2006). According to Eric Raymond, if the so-called open standard doesn’t have an open-source reference implementation, the term “standard” is an abuse of the language (Taft, 2004).

Based on the role that open source can play to make a good open standard better, Tiemann (2006) elaborates on Raymond’s statement above (Taft, 2004) and proposes distinguishing four levels of open standards: 0, 1, 2 and 3. Table 2-5 summarizes the open standard definitions given by Perens, Krechmer and Tiemann.
Table 2-5: Open standards definition from Perens, Krechmer and Tiemann

<table>
<thead>
<tr>
<th>Stream</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Open standard principles</strong> (Perens, <a href="http://perens.com">http://perens.com</a>)</td>
<td><strong>Availability:</strong> available for all to read and implement.</td>
</tr>
<tr>
<td></td>
<td><strong>Maximize End-User Choice:</strong> create a fair, competitive market for implementations of the standard. They do not lock the customer into a particular vendor or group.</td>
</tr>
<tr>
<td></td>
<td><strong>No Royalty:</strong> free for all to implement, with no royalty or fee. Certification of compliance by the standards organization may involve a fee.</td>
</tr>
<tr>
<td></td>
<td><strong>No Discrimination:</strong> open standards and the organizations that administer them do not favor one implementor over another for any reason other than the technical standards compliance of a vendor's implementation.</td>
</tr>
<tr>
<td></td>
<td><strong>Extension or Subset:</strong> Implementations of open standards may be extended, or offered in subset form. However, certification organizations may decline to certify subset implementations, and may place requirements upon extensions.</td>
</tr>
<tr>
<td></td>
<td><strong>Predatory Practices:</strong> may employ license terms that protect against subversion of the standard by embrace-and-extend tactics.</td>
</tr>
<tr>
<td><strong>Open standard requirements</strong> (Krechmer 2005)</td>
<td>Open Meeting, Consensus, Due Process, Open IPR, One World, Open Change, Open Documents, Open Interface, Open Access, On-going Support</td>
</tr>
<tr>
<td><strong>Four levels of open standards</strong> (Tiemann, 2006)</td>
<td><strong>Open Standard 0:</strong> The standard is documented and can be completely implemented, used, and distributed royalty free. Implementations of the standard may be extended, or offered in subset form.</td>
</tr>
<tr>
<td></td>
<td><strong>Open Standard 1:</strong> There is a specified OSS (Open source software) that can interoperate with the standard.</td>
</tr>
<tr>
<td></td>
<td><strong>Open Standard 2:</strong> There is a reference implementation that provides the ability to review the actual working of the standard. The reference implementation not only gives users of the standard a guaranteed exit strategy but also can be extremely valuable in identifying gaps or hidden assumptions that may underlie a standard definition. (This is the minimum level that Eric Raymond would endorse as a true open standard.)</td>
</tr>
<tr>
<td></td>
<td><strong>Open Standard 3:</strong> The reference implementation of the standard is an open source implementation that can be most useful in providing a means for i) advancing the standard over time as practices improve, and ii) providing a safeguard against fragmentation when a proprietary implementation extends the standard but the extensions have not been reincorporated into the open source reference implementation.</td>
</tr>
</tbody>
</table>

Existing research literature focuses predominantly on the specification (the specific content) of the standards and neglects the inherent role of the implementation of a standard in both defining and promulgating that standard (West, 2004). There is a very small body of formal theory related to the discussion of the relationship between a given standard and its specific implementation mechanisms. Although showing recent signs of advancement, the literature of standard adoption still seems fragmented. Most of the academic literature on standards is concerned with the economic analysis of network externalities and compatibility, the role of standards as tools of competitive strategy, and the determinants of the use and supply of standards (Bailetti & Callahan, 1995). A major reason is the lack of a consistent framework to unify various factors that may affect standard adoption.

2.4 Diffusion of innovations and technology adoption

2.4.1 Diffusion of innovation

The theoretical foundation of the technology adoption research is found in the diffusion of innovation literature (Rogers, 1983, 1995, 2003). Rogers proposed five major characteristics of innovations that influence whether and at which rate such innovations are adopted - relative advantage, compatibility, complexity, observability and trialability.
Table 2-6: Characteristics of a technology influencing its diffusion rate

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative advantage</td>
<td>Perceived benefits of a technology over previous technologies and the extent to which it is better than the idea it supersedes</td>
</tr>
<tr>
<td>Compatibility</td>
<td>The degree to which a technology and the tasks it performs are perceived as being consistent with the existing values, beliefs, past experiences, and needs of potential adopters</td>
</tr>
<tr>
<td>Complexity</td>
<td>Level of difficulty of installing and using a technology (e.g. variety and uncertainty increase complexity)</td>
</tr>
<tr>
<td>Observability</td>
<td>The degree to which the features and benefits of a technology are visible, noticeable and understandable to self/others, the results can be described to non-users</td>
</tr>
<tr>
<td>Trialability</td>
<td>The ability to experiment or try (on a limited basis) the technology before formally adopting it</td>
</tr>
</tbody>
</table>


a) Relative advantage is the degree to which an innovation is perceived as better than the idea it supersedes. In the case of standard adoption, the degree of relative advantage may be measured in economic terms such as opportunities for faster development, less maintenance, and cost saving, but strategic advantages and prestige for using an emerging standard are also important.

b) Compatibility is the degree to which an innovation is perceived as being consistent with the existing technologies and past experiences of potential adopters.

c) Complexity refers to the degree to which an innovation is perceived as difficult to understand and use. Reducing the complexity of standards is critical to the standards’ success.

d) Trialability is the degree to which an innovation may be experimented with at least on a limited basis. Standards that have high Trialability may be adopted.
more quickly. Through initial pilot projects, organizations can reduce the uncertainty and risk for major deployment of these standards.

e) *Observability* is defined as the degree to which the results of an innovation are visible to others.

For relative advantage, complexity and compatibility, the theoretical prediction from Roger’s model have found strong empirical support from numerous studies (Davis, 1989; Tornatzky, & Fleischer, 1990; Moore & Benbasat, 1991). Because empirical support has been weak for the remaining two characteristics observability and trialability are two arguable dimensions of theoretical prediction from Roger’s model (Moore & Benbasat, 1991).

### 2.4.2 Technology adoption life cycle

The Technology Adoption Life Cycle Model (Moore, 2000) is derived from Rogers’ innovation diffusion and clearly illustrates the evolution of a technology enabled market that develops in a characteristic pattern.

Moore’s key insight includes: 1) using how people respond to product and service innovations to explain that different adopter categories adopt innovations for different reasons and 2) identifying a gap or Chasm between the first two adopter groups
(innovators/early adopters) and the early majority for discontinuous or disruptive innovations while Rogers suggested smooth transitions between categories.

2.5 Lessons Learned

The following lessons were learned from the literature review.

The need of an open source development framework

There is a need of an analytical framework in which open source development is characterized in terms of generalized principles, as distinct from principles peculiar to software development (Hope, 2004).

User innovation network

User innovation network theory (von Hippel & von Krogh, 2001; von Hippel, 2002) describe the open source software development by the presence of five characteristics: free revealing, user innovation community, collective invention, commons-based peer production, user community support and governance

Open source project success measures

The number of developers, the number of downloads, the number of contributions, the number of new releases, the development status and maturity, and the achievement of preliminary identified goals could be used as success measures of open source software projects (Crowston, Annabi & Howison, 2003; Rehman, 2006).
Methodology for studying the development and evolution of open assets

Longitudinal field studies were suggested as a suitable methodology for studying the “life cycle” patterns of open source project evolution (von Hippel, 2002).

Open standard definition

Open Standards can be defined by means of a grading framework (Tiemann, 2006) involving four different levels; i) Level 0: the standard is documented and can be completely implemented, used, and distributed royalty free, however, the implementation is not certified; ii) Level 1: there is a specified open source software that can interoperate with the standard; iii) Level 2: there is a specified open source software that can explain and advance the standard as practices improve; iv) Level 3: This implementation of the standard is an open source implementation.

Lack of research on the relationship between open standards and the characteristics of open asset development

The definition of open standards involves the degree of “openness” of the reference implementation of these standards (West, 2004; Tiemann, 2006) However, existing research focuses on the evolution or adoption of standard specifications and neglects the relationship between the evolution of the open standards and the open source practices associated with their implementation (Simcoe, 2005; Tiemann, 2006).
is little research on the relationship between the specific standard implementation mechanisms and the standard’s adoption process (West, 2004).

**Longitudinal field research on change**

Longitudinal research on change should explore the context, content, and process of change together with their interconnections through time (Pettigrew, 1990, 2001).

Variables are not an one-to-one representation of the process itself and processes are closer to the actual behavior than are the variables (MacKenzie, 2000).
3 RESEARCH METHOD

This chapter is organized into three sections. Section 3.1 identifies the unit of analysis, section 3.2 discusses the study period, section 3.3 provides a very short summary of the research method, section 3.4 identifies the research steps and sections 3.5 to 3.12 briefly describe the research steps.

3.1 Unit of analysis

The unit of analysis is the OpenAccess project.

3.2 Study period

The study period is from the fourth quarter of 1999 to December 2006. The fourth quarter of 1999 was selected as the start date of data collection because it is the time when the Si2 launched the OpenAccess project. December 2006 was identified as an appropriate end of the research study period. This is the end of the quarter when the first commercial EDA design tool solely committed to the OA standard was introduced.

3.3 Research method

The research uses a combination of inductive and deductive approaches (Christensen & Raynor, 2003). Such an approach is a combination of two of the three reasons for using single cases provided by Siggelkow (2007): i) as an inspiration for new ideas (inductive stage), since there exist only limited theoretical knowledge concerning the
characteristics of open standard development project; and ii) as an illustration of the suggested conceptual framework (deductive phase) describing user innovations. In the inductive phase, archival and public source data about the OpenAccess project were first collected and a pure narrative story was developed following a temporal presentation format. Secondly, former observations were categorized into four different layers based on the insights from the longitudinal field research method (Pettigrew, 1990). Chronological timelines of events associated with each of the layers were then developed and graphically presented. Finally, the interdependence relationships between events and observations in different layers were discussed and insights formulated. The whole inductive phase resulted in a diagnostic case containing i) all of the features of the analytical chronologies, ii) a graphical representation and analysis of links between events in the four different layers.

In the next deductive phase, the available data was compared and related with the existing literature on user innovation networks, open source software, technology adoption and diffusion of innovation. Based on these linkages, the research insights were generated. A similar approach based on a combination of an inductive and a deductive phase was recently used by Sull (2005) to study the relationship between top-down investment and the failure of bottom-up strategic processes.

3.4 Research steps

The research method is comprised of the following steps:
• Collect data and prepare a case study of the OpenAccess project.
• Identify research questions.
• Identify layers and prepare a chronological timeline of events associated with each layer.
• Identify variables in the layers.
• Identify linkages between events and variables across layers.
• Identify constructs and their constituent dimensions.
• Examine the differences between the characteristics of the OA reference database implementation and the characteristics common to open source assets.
• Relate the answers to the research questions to existing literature.

3.5 Collecting data and preparing a case study

The history of the OA project is documented very well. The initial design of the OA project involved an open evolution process as a major part of the standard development strategy. There is a periodically updated OpenAccess™ Release 2.2 Standard API Tutorial (5 editions so far) published by the Si2 to familiarize readers with the history, evolution, and advantages of the OA project. The tutorial also describes the OA standard specification, functional capabilities, architecture and applications of existing EDA tools.

The archives of the Si2 provided another source of data for this study (see Table 3-1 for a summary of Si2 data sources). These records included detail transcripts of
comments, exhibitions, and discussions mainly presented in the bi-annual OpenAccess Conference, Si2 news releases, and the official websites of Si2 (www.si2.org and openeda.si2.org). These archives provided information for the analysis.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Period covered</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si2 news archives</td>
<td>Sep. 1999 - Dec. 2006</td>
<td>News about the status and activities of Si2 which were published on the <a href="http://www.Si2.org">www.Si2.org</a> in the “Si2 InTheNews” and “Si2 News Archives” column</td>
</tr>
<tr>
<td>Event reports</td>
<td>Feb. 2003 - Nov. 2006</td>
<td>Event report from the OA Conferences, the OA Panel at the industry conferences, OA developers’ forum, and Interoperability Workshops</td>
</tr>
<tr>
<td>OpenAccess Book</td>
<td>2003 - 2006</td>
<td>OpenAccess: the Standard API Tutorial serials book which were published by Si2 (OpenEDA.Si2.org) with the main release of OA standard</td>
</tr>
<tr>
<td>OA release notes and What’s new</td>
<td>2002 - 2006</td>
<td>“Release notes” are the official release notes published by Si2 and “what’s new” document was released together with each code package</td>
</tr>
<tr>
<td>OA community activities</td>
<td>Dec.2003 - Dec.2006</td>
<td>The community activities such as: downloads, report bug which were recorded in the OpenEDA.Si2.org by the release tracker, forum, etc.</td>
</tr>
<tr>
<td>Si2 by-laws</td>
<td>1999 - 2006</td>
<td>“Amended and Restated By-laws of Silicon Integration Initiative, Inc.” published by the Si2</td>
</tr>
</tbody>
</table>

Using the Goggle search engine an online search was undertaken using “OpenAcess” and “Open Access” as the keywords. A similar search was undertaken for each of the following websites:

- EE-times, EDAnacle, Demosondemand (major industry newspapers)
- EDN, Electronic Design, DeepChip (technical journal publications)
- company websites of OpenAccess Coalition members
Using the data collected a descriptive single case study of the OA project was written. The case study is included in Appendix A.

3.6 Identify research questions

Three research questions were identified inline with the formulated research objective:

• What are the multidimensional constructs that could be used to describe the development of the OpenAccess standard?

• What are the differences between the characteristics of the OA reference database implementation and the characteristics common to open source assets?

• What is the relationship between the constructs used to describe the OpenAccess standard and the research literature?

3.7 Identification of four layers

The case study provided in Appendix A was used as a data repository for this research. The longitudinal field research methodology suggests using multiple contextual layers of events to study the temporal dynamics of change. In our case, the events that were found to be relevant for the development of the OpenAcess standard were organized into four layers: 1) Si2/OAC decisions, 2) OA members’ activity, 3) OA RDBI, and 4) OA API specification, and presented in a chronological timeline.
The identification of the layers was driven by the need to describe the i) context (layer 1 & 2), ii) process (layer 3), and iii) content (layer 4) of the standard development process.

Layer 1 (Si2/OAC decisions) and 2 (OA members’ activity) characterize the proactive agents – the standard setting organization and the industrial community - interested in the development and the adoption of the standard. The activities of these agents manifest the contextual aspects of the standard development and adoption process – the various technological, economical and business conditions, motivations and interests of the different players within the EDA, semiconductor and electronic systems design industries.

This particular choice of the first two layers is in accordance with the main criterion for context level selection which is used in longitudinal field research – the inseparable role of context and action. The analysis of the events in these two layers is expected to demonstrate how the various aspects of the OA project design, structure and overall industrial context are mobilized by the different actors and groups as they seek to obtain outcomes important to them (Pettigrew, 1990). While the Si2/OAC decision layer describes the overall industrial context expressed through the commitment of the Si2 to address the EDA design tool interoperability issues from a more global perspective, the OA members’ activity layer describes the standard adoption context from an individual company perspective.
The third layer was the OA reference database implementation (RDBI) layer. The choice of this layer was motivated by the fact that: i) all changes of the OA standard specification must be suggested through the RDBI, and ii) the open access, continuous improvement, and benefits from using the RDBI provide the practical motivations for companies to adopt the OA standard. The change events within the RDBI layer characterize the process aspects of the OA project.

The main objective of the OA project is set to develop and enable the adoption of the OA API. The OA standard specification is the only part that needs to be standardized, i.e., the OA API specification layer represents the content aspect of the OA project.

3.8 Identification of variables

No variables were measured for layer 1 – Si2/OAC decisions.

The variables describing changes in layer 2 (OA members' activities) are summarized in Table 3-2. For each variable, Table 3-2 identifies the interval for data collection, study period and describes how data was measured.
### Table 3-2: Variables in the OA members' activity layer

<table>
<thead>
<tr>
<th>No.</th>
<th>Variable Description</th>
<th>Data interval</th>
<th>Study period</th>
<th>What was measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of Si2 corporate members</td>
<td>Collected annually</td>
<td>2000-2006</td>
<td>The number of Si2 corporate members who have signed the OA license according to company’s category</td>
</tr>
<tr>
<td>2</td>
<td>Number of OAC member companies</td>
<td>Collected annually</td>
<td>2000-2006</td>
<td>Total number of OAC member companies according to company’s category</td>
</tr>
<tr>
<td>3</td>
<td>Number of OA EDA design tools</td>
<td>Collected annually</td>
<td>2000-2006</td>
<td>Total number of design tools complying to the OA standard which including 1. Commercial EDA design tools: the number of announced commercial EDA tools providing OA interface 2. In-house EDA design tools: the number of built in-house tools based on OA interface for internal use</td>
</tr>
</tbody>
</table>

### Table 3-3: Motives and risks for companies to participate in the OA project

<table>
<thead>
<tr>
<th>No.</th>
<th>Motives and risks Description</th>
<th>Data sub-sample</th>
<th>What was examined</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Motives and risks of the technology provider when releasing its technology as part of the open standard</td>
<td>Cadence</td>
<td>The motives and risks for Cadence to release its Genesis reference database implementation as version 1.0 of the OA standard</td>
</tr>
<tr>
<td>2</td>
<td>Motives and risks of adopters of the OA standard</td>
<td>Sub-sample of 12 OAC member companies (randomly selected within each company category)</td>
<td>The motives and risks of OAC members to participate in the OA community and comply their products with the OA standard</td>
</tr>
</tbody>
</table>

To identify motives and risks of companies participating in the OA project, key words were first extracted from the text data. Then these key words were categorized and summarized. Table 3-3 shows the two categories of motives and risks: i) of the
technology provider (Cadence), and ii) of OAC members. Table 3-4 describes the company categories used in this study.

<table>
<thead>
<tr>
<th>Company category</th>
<th>Criteria</th>
<th>Identification keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 SEMI &amp; System</td>
<td>Main products are electronic product or IC design solution including silicon intellectual property</td>
<td>Manufacture, equipment, device, hardware, foundry, design service, pack, IP, SIP, semiconductor, fab, fabless, IDM (int. device manufacturer)</td>
</tr>
<tr>
<td>2 EDA companies</td>
<td>* Main products are electronic design automation (EDA), software, tools, methodology, analysis * non-start-up (mature) companies</td>
<td>Electronic design automation (EDA), software, tools, methodology, analysis, IC layout</td>
</tr>
<tr>
<td>3 EDA start-ups</td>
<td>* EDA company founded less than 3 years when they joined OAC or Si2 * Funding holder is private and usually invested by venture capital</td>
<td></td>
</tr>
</tbody>
</table>

The variables used to measure change in layer 3 (OA RDBI) are given in Table 3-5.

<table>
<thead>
<tr>
<th>No.</th>
<th>Variable</th>
<th>Data interval</th>
<th>Study period</th>
<th>What was measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of RDBI source or binary code releases</td>
<td>Collected annually</td>
<td>2001-2006</td>
<td>Total number of new OA RDBI released per year including both releases to members only and general releases to the public</td>
</tr>
<tr>
<td>2</td>
<td>Number of RDBI downloads</td>
<td>Collected annually</td>
<td>2001-2006</td>
<td>Total number of OA RDBI package downloads per year</td>
</tr>
<tr>
<td>3</td>
<td>Number of RDBI functionality changes</td>
<td>Collected quarterly</td>
<td>2001-2006</td>
<td>Total number of functionality changes based on its previous version including the new functionality changes and items fixed²</td>
</tr>
<tr>
<td>4</td>
<td>Number of RDBI contribution records</td>
<td>Collected quarterly</td>
<td>2004-2006³</td>
<td>Total number of records in the OA track (including error reports and new feature requests)</td>
</tr>
</tbody>
</table>

² The statistical data is provided in the OA standard release notes
³ Due to the system upgrade of www.openeda.org, all OA track data before December 2003 are not available any more. The study period of contribution records from OA community had to begin from 01/01/2004
The variables used to measure changes in layer 4 (OA API) are summarized in Table 3-6.

### Table 3-6: Variables in the OA API layer

<table>
<thead>
<tr>
<th>No.</th>
<th>Variable</th>
<th>Data interval</th>
<th>Study period</th>
<th>What was measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of API releases</td>
<td>Collected annually</td>
<td>2001-2006</td>
<td>Total number of new “OpenAccess C++ API Documentation” per year including both releases to OAC members only and general releases to the public</td>
</tr>
<tr>
<td>2</td>
<td>Number of API specs downloads</td>
<td>Collected annually</td>
<td>2001-2006</td>
<td>Total number of “OpenAccess C++ API Documentation” downloads per year</td>
</tr>
<tr>
<td>3</td>
<td>Number of API changes</td>
<td>Collected quarterly</td>
<td>2001-2006</td>
<td>Total number of API changes(^4) contained in the last official OA API specification release as compared with the previous one.</td>
</tr>
</tbody>
</table>

#### 3.9 Identification of linkages between events and variables

The analysis of the relationship between events and variables across layers is based on the insights from the longitudinal field research method:

- focus on context, content, and process of change together with their interconnections through time (Pettigrew, 1990)
- a source of change is the asymmetry between levels of context, where processes at different levels of analysis are often observed to have their own momentum, rates, pace and trajectory (Pettigrew, 1990; Pettigrew et al. 2001).
- new events are triggered by past events, i.e. previous events shape future behavior (Kimberly & Bouchikhi, 1995).

---

\(^4\) The sum of the number of API changed, the number of Class changed, and the number of Enums changed, which are noted by the OA standard release notes.
• questions such as what, who, where, when, why, and how are the special features of a process vocabulary that can help uncover novel theoretical questions and reveal original findings (Pettigrew et al., 2001)

• the change process is understood as movement from one state to another

(Weick & Quinn, 1999) and should be encapsulated by appropriate variables.

3.10 Identification of constructs

The multidimensional constructs and their constituent dimensions were drawn from the information on the case study. Particular attention was paid to the linkages between events and variables across layers. The main criteria for the selection of the constructs were again based on longitudinal field research insights focusing on the necessity to account for the content, process, context and the agents of change.

3.11 Differences between the OA RDBI and open source assets

To examine the differences between the characteristics of the OA RDBI and the characteristics common to open source assets, the researcher used the open source definition provided by the Open Source Initiative (www.opensource.org).

3.12 Relate the answers to the research questions to existing literature

The answers to the research questions were analyzed within the context of four different perspectives: open standards, open source software, Rogers’ diffusion of innovation theory and Moore’s technology adoption life cycles.
4 RESULTS

This chapter is organized into three sections. Section 4.1 provides a brief introduction of the OA project. Fuller details on the OA project can be found in the case study which is included in Appendix A of this thesis. Sections 4.2 to 4.5 describe the chronological timeline of subsequent events that happened in the four different contextual layers. Section 4.6 contains a cross-layer analysis.

4.1 A short description of the OpenAccess (OA) project based on the case study

It is widely accepted that the lack of interoperability between EDA design tools is a major limitation for the cost-effective design and manufacturing of silicon integrated circuits. To solve this problem, the semiconductor, electronic systems, and EDA industries initiated a variety of efforts to standardize the EDA design data format. The OA project was founded by Silicon Integration Initiative (Si2) in late 1999 to provide an industry-accepted API-based design data format standard.

4.1.1 Project objective

The Si2 positioned the OA project as a community effort to provide true interoperability, not just data exchange, among IC design tools through the development and the adoption of an open standard design data API and reference database supporting that API. The main goal of OA project was set to “gain adoption of the standard within the EDA industry.”
4.1.2 Three building blocks of the OA project

The OA project comprises three building blocks

A. OA standard specification

The OA standard specification includes three components defining the API:

- an information model defined by a collection of entity relationship diagrams and describing a conceptual perspective of the objects and their mutual relationship including the navigability across object relationships
- a data model defined by C++ header files, which specify software class and function interface details
- the API specifications which presents the header information in a more readable format

B. Reference database implementation (RDBI)

The OA reference database implementation (RDBI) is the software implementing the OpenAccess API standard. The OA RDBI was designed to be developed as just one particular implementation of the API, with publicly available source, updated in sync with any changes to the API standard.

C. OpenEvolution process

The OpenEvolution process is the governance process managing the on-going development, distribution, and support of the OA standard specification and RDBI.
4.1.3 OA governance structure

At the heart of OA project governance structure are several groups of stakeholders (Figure 4-1) controlling the evolution of the OA specification standard in terms of funding as well as technical guidance including:

- the OpenAccess Coalition (OAC) members – a body composed by any company or bona fide industry organization who commits in contributing engineering and financial resources to the development, evaluation, and adoption of OA specification standard
- the Change Team members – a body of technical experts who serve the OAC to manage the evolution of OA specification standard
• the Integrator - Cadence serve as the integrator of the OpenAccess RDBI
• the Working Groups – bodies enabling the joint cooperation of many different professionals with expertise anchored around a specified development task

Surrounding this core of active participants is the OpenAccess Community including anyone who wishes to use or contribute to OA. A broad OA community base of talent and resources provide the necessary mechanisms for downloading the OA specification standard and RDBI, logging bug reports and enhancement requests, or simply asking questions in a familiar open forum environment.

4.1.4 RDBI development process

If the user makes enhancements to RDBI and wishes them to be part of the OA standard (including both – the OA standard specification and the RDBI), then the changes are submitted back to the Change Team. The Change Team considers all such requests and, if accepted, forwards them to the OA Integrator for incorporation into the OA standard. Accepted changes may be incorporated as submitted by the user or modified by the Integrator for consistency with the OA architecture. A working group may be convened for further study to insure sufficient generality in the final solution.

If a new feature is requested without accompanying implementation code, the OAC may consider funding or a request for technology (RFT) to assist in the development.
Approved enhancements are published in a preliminary developed Roadmap along with an approximate schedule. Once a contribution is adopted or a change request is implemented, it is incorporated into a OA standard release and published on the official OA website. Even if a requested enhancement is ultimately rejected during the Change Team review process the Open-Access extensibility features can usually be used to accomplish the desired result for the user, without resorting to a change in the OA core.

4.1.5 OA license terms

Table 4-1 provides a summary of OA license terms.

<table>
<thead>
<tr>
<th>Membership</th>
<th>Non exclusive, royalty free, worldwide, copyright license to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Si2 member</td>
<td>Use, reproduce and prepare derivative works of the software contained in the Package and the API specification for such software in both source and binary code forms solely for Non-commercial Use</td>
</tr>
<tr>
<td>Si2 academic member</td>
<td>Use, reproduce and prepare derivative works of the software contained in the Package and the API specification for such software in both source and binary code forms solely for Non-commercial Use within the scope of member's academic activities</td>
</tr>
<tr>
<td>Si2 corporate member</td>
<td>Use, reproduce and prepare derivative works of the software contained in the Package and the API specification for such software in both source and binary code forms solely for Internal Use</td>
</tr>
<tr>
<td>OAC member</td>
<td>Reproduce, distribute and sublicense the unmodified software contained in the Package in binary code form only and the unmodified documentation contained in the Package, and to include in any such distribution bona fide error corrections to the software in binary code form only and documentation contained in the Package that do not alter the syntax, semantics and/or behavior of the API specification for such software, solely in connection with the offerings that Recipient markets as part of its product and services offerings in the normal course of its business, which licenses shall remain in effect in perpetuity unless terminated in accordance with this Agreement.</td>
</tr>
</tbody>
</table>

Source: OpenAccess – Internal Use and Distribution License V 4.0 (with permission of the Si2)
4.2 Chronological timeline of events in the Si2/OAC decisions layer

This layer of events includes the governance or management decisions of the OAC of the Si2 to influence the development of the OA standard. Figure 4-2 and Table 4-2 summarize the temporal development of events in this layer.

Figure 4-2: Chronological timeline of events in the Si2/OAC decision layer

Table 4-2: Description of the events in the Si2/OAC decision layer

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Date</th>
<th>Event</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1,1</td>
<td>1999, Q4</td>
<td>Silicon Integration Initiative (Si2) founded the OpenAccess project and the OpenAccess Coalition (OAC)</td>
<td><a href="http://www.si2.org/?page=76">http://www.si2.org/?page=76</a></td>
</tr>
<tr>
<td>E1,2</td>
<td>1999, Q4</td>
<td>OAC issued a request-for-technology (RFT) looking for a proven API specification and RDBI that could be commercially adopted and released as open source</td>
<td>OA book (2006), pp.4</td>
</tr>
<tr>
<td>E1,3</td>
<td>2000, June</td>
<td>Si2 launched OpenEDA.org and to move interoperability from committees to communities</td>
<td><a href="http://www.eetimes.com/story/OEG20000604S0002">http://www.eetimes.com/story/OEG20000604S0002</a></td>
</tr>
<tr>
<td>E1,4</td>
<td>2000 Sep.</td>
<td>OAC accepted Cadence’s response to the RFT providing freely their Genesis API specs to the OAC</td>
<td><a href="http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm">http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm</a></td>
</tr>
<tr>
<td>E1,5</td>
<td>2001, June 18</td>
<td>OAC reached a community-source agreement with Cadence to use their Genesis RDBI source code as the basis of the OA standard</td>
<td><a href="http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm">http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm</a></td>
</tr>
</tbody>
</table>
4.3 Chronological timeline of events in the OA members' activity layer

This layer includes the events associated with the activities of the OA community members (e.g., joining the OAC, downloading the RDBI code, suggesting modifications to the API, etc.). Figure 4-3 and Table 4-3 summarize the temporal development of events in this layer.

Figure 4-3: Chronological timeline of events in the OA members' activity layer

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Event ID</th>
<th>Date</th>
<th>Event</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2,1</td>
<td>1999, Q4</td>
<td>Cadence(^5) and Mentor Graphics joined OAC as founding members but not Synopsys (second in revenue)</td>
<td>2000 OAC membership list from Si2</td>
</tr>
<tr>
<td>E2,2</td>
<td>2000 June</td>
<td>Synopsys expresses its openness to join OAC (announced at DAC)</td>
<td><a href="http://www.eetimes.com/story/OEG20020204S0022">http://www.eetimes.com/story/OEG20020204S0022</a></td>
</tr>
<tr>
<td>E2,3</td>
<td>2000 Sep.</td>
<td>Cadence responded to the OAC RFT and contributed its Genesis API specs to OAC</td>
<td><a href="http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm">http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm</a></td>
</tr>
<tr>
<td>E2,4</td>
<td>2001 June 18</td>
<td>Cadence agreed to contribute Genesis DBI source code (no release yet)</td>
<td><a href="http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm">http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm</a></td>
</tr>
<tr>
<td>E2,5</td>
<td>2001 June</td>
<td>Cadence contributed Genesis binary (only) code to OAC</td>
<td>Release notes</td>
</tr>
<tr>
<td>E2,6</td>
<td>2001 June 28</td>
<td>First downloads of the OA1.0 API specs and RDBI binary code from the public</td>
<td>Data provided by Si2</td>
</tr>
<tr>
<td>E2,7</td>
<td>2002 Feb. 4</td>
<td>Cadence acquired veto power on OA API specs change decisions and released Genesis source code (OA 1.0) to the OAC.</td>
<td><a href="http://www.cadence.co.jp/news/h14-2-4_c.html">http://www.cadence.co.jp/news/h14-2-4_c.html</a></td>
</tr>
</tbody>
</table>

\(^5\) Appendix B shows top five EDA companies by market value 2005 (US$)
<table>
<thead>
<tr>
<th>E2,14</th>
<th>2003 Sep.23</th>
<th>Mentor joins back the OpenAccess Coalition as part of the settlement of lawsuits with Cadence</th>
<th><a href="http://www.eetimes.com/news/design/index.jhtml?sessionid=KU2JR2JKJU5PUQSNLOSKHSCJUNN2JVN">http://www.eetimes.com/news/design/index.jhtml?sessionid=KU2JR2JKJU5PUQSNLOSKHSCJUNN2JVN</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>E2,17</td>
<td>2004 July</td>
<td>Cadence handed the veto power to the OAC change team</td>
<td><a href="http://www.eetimes.com/story/OEG20020204S0022">http://www.eetimes.com/story/OEG20020204S0022</a> <a href="http://www.si2.org/?page=85">http://www.si2.org/?page=85</a></td>
</tr>
<tr>
<td>E2,18</td>
<td>2004 Nov.</td>
<td>Startup SiNavigator created an OA new business model</td>
<td><a href="http://www.si2.org/?page=261">http://www.si2.org/?page=261</a></td>
</tr>
<tr>
<td>E2,19</td>
<td>2005 June</td>
<td>Synopsys joined OAC at director level &amp; began promoting OA</td>
<td><a href="http://www.si2.org/?page=483">http://www.si2.org/?page=483</a></td>
</tr>
<tr>
<td>E2,20</td>
<td>2006 Sep.</td>
<td>Last version of Cadence’s Virtuoso supports OA only</td>
<td><a href="http://www.edadesignline.com/products/192700790">http://www.edadesignline.com/products/192700790</a></td>
</tr>
</tbody>
</table>

### 4.3.1 Si2/OAC membership

Figure 4-4 provides i) the number of Si2 members per year who have signed the OA license (continuous curve with diamond-like labels) and ii) the growth of OAC membership. The number of OAC member companies per year is represented by three color columns. A different gray scale color scheme was used to illustrate the relative distribution of the three company categories: semiconductor & systems companies, mature EDA companies, and EDA start-ups.
The analysis of Figure 4-4 leads to the following conclusions.

- Both Si2 and OAC membership kept increasing over the study period. Starting with 8 members (the OAC founding members) in 2000, the number of Si2 members reached 99 in 2006. The OAC membership increased approximately 4 times and included the top 5 EDA companies in the world (Appendix B).

- Two periods in the membership growth history can be identified: i) a slow growth period (2000 to 2003); ii) a fast growth period (2003 to 2006). The two periods can be clearly identified by the different slope values of the curve describing the growth of the Si2 membership.

- In the slow growth period the majority of OAC members were mostly end-user companies (semiconductor and electronics systems). In this initial period new members from the EDA industry were rare. The first EDA start-ups started joining in 2003 and their number grew dramatically in 2004. Beginning from 2003 the end-user membership stopped growing and the EDA companies became the main
source of OAC membership growth.

4.3.2 Number of OA EDA design tools

Figure 4-5 provides the number of OA design tools developed per year as well as their distribution by company category. Again, two periods can be identified in the development of OA EDA design tools: an initial period when no OA tools were available, and a growth period when OA tools began to appear.

The initial period comprises the period from 2000 to 2003 – the same as the period of slow membership growth. During this period the main activities of Si2 members were focused on testing, evaluating and developing working familiarity with the OA API standard. There were no records of commercially released or in-house developed OA EDA design tools.

Figure 4-5: Newly announced commercial and in-house OA EDA tools per year

![Graph showing the number of newly announced commercial and in-house OA EDA tools per year from 2000 to 2006. The graph shows two categories: SEMI & System and EDA-mature, with EDA-startup as a separate category. The data is color-coded for each category.]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The growth period comprises the time from 2003 to 2006 – the same as the period of fast membership growth. OA-based EDA design tools started appearing on the market since 2003 from all three types of companies. The number of in-house OA design tools grew quickly in the initial 2 years of this period but the increase tendency became slow in 2005 and stopped in 2006. This could be an indication of the high level of maturity that was reached by the commercial EDA design tools supporting the OA standard.

Table 4-4 provides a summary of Si2/OAC membership and OA EDA design tools.

<table>
<thead>
<tr>
<th>Year</th>
<th>Si2 memb</th>
<th>Membership of OAC</th>
<th>Si2 member with announced OA tools</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>SEMI &amp; System</td>
</tr>
<tr>
<td>2000</td>
<td>8</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>2001</td>
<td>10</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>2002</td>
<td>19</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>2003</td>
<td>26</td>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>2004</td>
<td>49</td>
<td>27</td>
<td>11</td>
</tr>
<tr>
<td>2005</td>
<td>75</td>
<td>30</td>
<td>11</td>
</tr>
<tr>
<td>2006</td>
<td>99</td>
<td>37</td>
<td>11</td>
</tr>
</tbody>
</table>

4.3.3 Cadence’s motives for and risks from releasing its technology

The publicly announced motives of Cadence to release the Genesis API specification and RDBI include: (i) the desire to help and influence the development of the OpenAccess standard, (ii) the desire to respond to customers’ requests, (iii) the desire to increase the number of experienced developers working on the development of the asset, and (iv) the desire to increase the likelihood that competitors will contribute to
the RDBI code development. The vested interested of Cadence, as the major EDA
design tool vendor, to promote its own technology solution as part of the emerging OA
standard can be identified as another (although not publicly articulated) motive.

The major risk to which Cadence exposed itself was that the release of its technology
asset (the Genesis RDBI) led to a loss of competitive differentiation. An example of
such loss is based on the fact that Ciranova (Cadence’s competitor in the analog market)
used the OA RDBI for the development of its PyCell Studio product which allows any
vendor to access and generate PCell design layouts via the OA standard. The PCell
layout design tools count for a third of Cadence’s quarterly revenue. A key reason for
Cadence to be able to hold onto that share over the years is its proprietary SKILL PCell
description format that is used to generate the PCells. The commercial release of
Ciranova’s PyCell Studio resulted in Cadence’s loss of competitive advantage.

4.3.4 Motives and risks for companies adopting the OA standard

Table 4-5 illustrates companies’ motives to participate in the OA project based on
news releases from a sub-sample of 12 OAC members.

A common motive referred by nearly all companies is the realization of tight
integration and high interoperability by the standardized API. Other motives include:
fast quality improvement (for semiconductor and system companies), lowering the
market barrier to allow concentration on core competence and fast quality improvement at low cost (for mature EDA companies), and freely adding new features (for EDA startups).

### Table 4-5: OAC members’ motives to adopt the OA standard

<table>
<thead>
<tr>
<th>Motive</th>
<th>SEMI&amp; System</th>
<th>EDA</th>
<th>EDA-startups</th>
<th>Total</th>
<th>Count</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tight integration and high interoperability by the standardization</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>11</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>Lower market barrier</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>42%</td>
<td></td>
</tr>
<tr>
<td>Obtaining indirect revenues by selling related products</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Better relations with customers</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>17%</td>
<td></td>
</tr>
<tr>
<td>Low cost and fast quality improvement</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>67%</td>
<td></td>
</tr>
<tr>
<td>Flexibility to add new features</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>Collective protection of technological advances and avoiding lock-in</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>17%</td>
<td></td>
</tr>
</tbody>
</table>

The data analysis identified three risks including: (i) the lack of OA interface from main EDA design tools (for semiconductor and system companies), (ii) the limitation in functionality (for mature EDA companies), and (iii) the poor documentation at the initial stage (for mature EDA companies). There was no data about risks associated with EDA start-ups.

### 4.4 Chronological timeline of events in OA RDBI layer

Layer 3 refers to the events associated with the OA RDBI including source or binary code releases published by the OAC. Figure 4-6 and Table 4-6 summarize the temporal development and description of the events in this layer.
Figure 4-6: Chronological timeline of events in the OA RDBI layer

Table 4-6: Description of the events in the OA RDBI layer

<table>
<thead>
<tr>
<th>ID</th>
<th>Date</th>
<th>Event</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>E3,1</td>
<td>2001 June</td>
<td>Genesis RDBI binary code open to Si2/OAC</td>
<td><a href="http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm">http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm</a></td>
</tr>
<tr>
<td>E3,2</td>
<td>2001 June 28</td>
<td>Genesis RDBI binary code released to the public as OA 1.0 RDBI binary code</td>
<td><a href="http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm">http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm</a></td>
</tr>
<tr>
<td>E3,3</td>
<td>2002 Feb. 4</td>
<td>Genesis RDBI source code open to the public</td>
<td><a href="http://www.eetimes.com/story/OEG20020603S0047">http://www.eetimes.com/story/OEG20020603S0047</a></td>
</tr>
<tr>
<td>E3,4</td>
<td>2002 May</td>
<td>OA 2.0 RDBI source code open to the Si2/OAC (the completely re-written in C++ Genesis code)</td>
<td><a href="http://www.eetimes.com/story/OEG20020603S0047">http://www.eetimes.com/story/OEG20020603S0047</a></td>
</tr>
<tr>
<td>E3,6</td>
<td>2003 May 12</td>
<td>OA 2.1 RDBI source code open to the Si2/OAC (enables production)</td>
<td><a href="http://www.si2.org/?page=385">http://www.si2.org/?page=385</a></td>
</tr>
<tr>
<td>E3,7</td>
<td>2003 July</td>
<td>OA 2.1 RDBI source code released to the public</td>
<td><a href="http://www.si2.org/?page=382">http://www.si2.org/?page=382</a></td>
</tr>
<tr>
<td>E3,8</td>
<td>2004 Nov.</td>
<td>OA 2.2 RDBI source code released to Si2/OAC (includes OA/Milkyway translation)</td>
<td><a href="http://openeda.si2.org/si2_online/ReleaseNotes/">http://openeda.si2.org/si2_online/ReleaseNotes/</a></td>
</tr>
<tr>
<td>E3,9</td>
<td>2004 Nov.</td>
<td>OA 2.2 RDBI source code released to the public</td>
<td><a href="http://openeda.si2.org/si2_online/ReleaseNotes/">http://openeda.si2.org/si2_online/ReleaseNotes/</a></td>
</tr>
</tbody>
</table>
Table 4-7: Descriptive statistics of the variables associated the OA RDBI layer

<table>
<thead>
<tr>
<th>Year</th>
<th>Number of Si2/OAC member releases</th>
<th>Number of general releases</th>
<th>Number of RDBI func. changes</th>
<th>Number of RDBI downloads</th>
<th>Number of RDBI contribution records</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bug reports No data available</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>1 (binaries)</td>
<td>1 (binaries)</td>
<td>0</td>
<td>497</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>2 (source)</td>
<td>1 (source)</td>
<td>0</td>
<td>1904</td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>6</td>
<td>4</td>
<td>212</td>
<td>4868</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>7</td>
<td>1</td>
<td>95</td>
<td>2732</td>
<td>124 13</td>
</tr>
<tr>
<td>2005</td>
<td>8</td>
<td>2</td>
<td>593</td>
<td>3774</td>
<td>114 19</td>
</tr>
<tr>
<td>2006</td>
<td>5</td>
<td>1</td>
<td>421</td>
<td>4776</td>
<td>123 31</td>
</tr>
<tr>
<td>Total</td>
<td>28</td>
<td>9</td>
<td>1321</td>
<td>18551</td>
<td>424</td>
</tr>
</tbody>
</table>

Table 4-7 provides a descriptive statistics of the variables associated with the OA RDBI layer. The first RDBI releases were in 2001 and in binary form only. Since 2002 all releases were in source code form. Again, two periods in the RDBI development can be identified: i) an initial (no development) period from 2000 to 2003, and ii) a second (development) period from 2003 to 2006. The major event in 2003 enabling the RDBI development period was the first OA source code release to the public – OA version 2.0. It enabled the possibility for RDBI enhancement contributions and functionality changes as well as a dramatic growth in the number of RDBI source code downloads.

4.5 Chronological timeline of events in OA API layer

The OA API standard specifications represent the actual content of the standard which is intended to provide a standardized design format semantics and interfaces for
multi-vendor EDA design applications. Layer 4 refers to the events associated with the evolution of the OA standard specification. Figure 4-7 and Table 4-8 summarize the temporal development and description of the events in this layer.

Figure 4-7: Chronological timeline of events in the OA API layer

Table 4-8: Description of the events in the OA API layer

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Date</th>
<th>Event</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>E4,2</td>
<td>2001 June 28</td>
<td>Genesis API specification released to the public as OA 1.0 API specs</td>
<td><a href="http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm">http://archives.si2.org/Inthenews/pr2001/pr_061801_openaccess.htm</a></td>
</tr>
<tr>
<td>E4,3</td>
<td>2002 Feb.</td>
<td>OA 2.0 API specs open to the public</td>
<td><a href="http://www.si2.org/?page=292">http://www.si2.org/?page=292</a></td>
</tr>
<tr>
<td>E4,4</td>
<td>2002 March</td>
<td>OA 2.0 API specs released to the public</td>
<td><a href="http://www.si2.org/events_dir/2003/oaforum/oac.pdf">http://www.si2.org/events_dir/2003/oaforum/oac.pdf</a></td>
</tr>
<tr>
<td>E4,5</td>
<td>2003 May 12</td>
<td>OA 2.1 API specs open to the public</td>
<td><a href="http://www.si2.org/?page=385">http://www.si2.org/?page=385</a></td>
</tr>
<tr>
<td>E4,6</td>
<td>2003 July</td>
<td>OA 2.1 API specs released to the public</td>
<td><a href="http://www.si2.org/?page=382">http://www.si2.org/?page=382</a></td>
</tr>
<tr>
<td>E4,7</td>
<td>2004 Nov.</td>
<td>OA 2.2 API specs open to the Si2/OAC members only</td>
<td><a href="http://openeda.si2.org/si2_online/ReleaseNotes/">http://openeda.si2.org/si2_online/ReleaseNotes/</a></td>
</tr>
<tr>
<td>E4,8</td>
<td>2005 July</td>
<td>OA 2.2 API specs open to the public</td>
<td></td>
</tr>
</tbody>
</table>
Table 4-9: Descriptive statistics in OA API specification layer

<table>
<thead>
<tr>
<th>Year</th>
<th>Number of Si2/OAC member release</th>
<th>Number of General release</th>
<th>Number of API changes</th>
<th>Number of API downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2001</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>645</td>
</tr>
<tr>
<td>2002</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1258</td>
</tr>
<tr>
<td>2003</td>
<td>6</td>
<td>3</td>
<td>294</td>
<td>1355</td>
</tr>
<tr>
<td>2004</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1502</td>
</tr>
<tr>
<td>2005</td>
<td>5</td>
<td>2</td>
<td>131</td>
<td>1395</td>
</tr>
<tr>
<td>2006</td>
<td>3</td>
<td>1</td>
<td>257</td>
<td>2313</td>
</tr>
<tr>
<td>Total</td>
<td>18</td>
<td>9</td>
<td>640</td>
<td>8468</td>
</tr>
</tbody>
</table>

Table 4-9 provides the descriptive statistics of the variables associated with the OA API specification layer. The two periods associated with the development of the RDBI can be also identified in the case of the OA API development and evolution i) an initial (no development) period from 2000 to 2003, and ii) a second (development) period from 2003 to 2006. The beginning of the API development period is illustrated by the number of API changes that started in 2003. The OA API development was enabled by the possibility for the OA community members to use the RDBI (after the release of the OA 2.0 source code) as API change mechanism.
4.6 Cross-layer analysis

4.6.1 Linkages between events across the four layers

To identify links between events, a chronological timeline including the 4 contextual layers was developed (Figure 4-8).

The events in each of the layers presented in Figure 4-8 use the same labels and are presented in the same order as in Figures 4-2 (for layer 1), 4-3 (for layer 2), 4-6 (for layer 3) and 4-7 (for layer 4). The description of the events in each layer is given in Tables 4-2 (for layer 1), 4-3 (for layer 2), 4-6 (for layer 3) and 4-8 (for layer 4). Some of the events happened simultaneously. Two types of simultaneous events were identified – with and without association (Figure 4-9). Simultaneous events without association happened in the same moment of time by accident only but are not related to each other (for example by one triggering the other).
The OA project was launched without any existing API specification. This initiated a series of events with the following logic.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1,2</td>
<td>1999, Q4</td>
<td>OAC issued a request-for-technology (RFT) looking for a proven API specification and RDBI that could be commercially adopted and released as open source</td>
</tr>
<tr>
<td>E2,3</td>
<td>2000 Sep.</td>
<td>Cadence responded to the OAC RFT and contributed its Genesis API specs to OAC</td>
</tr>
<tr>
<td>E1,4</td>
<td>2000 Sep.</td>
<td>OAC accepted Cadence's response to the RFT providing freely their Genesis API specs to the OAC</td>
</tr>
<tr>
<td>E4,1</td>
<td>2000 Sep.</td>
<td>Cadence contributes its Genesis API specs to Si2/OAC</td>
</tr>
<tr>
<td>E1,5</td>
<td>2001, June 18</td>
<td>OAC reached a community-source agreement with Cadence to use their Genesis RDBI source code as the basis of the OA standard</td>
</tr>
<tr>
<td>E2,4</td>
<td>2001 June 18</td>
<td>Cadence agreed to contribute Genesis DBI source code (no release)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>E2,5</td>
<td>2001 June</td>
<td>Cadence contributed Genesis binary (only) code to OAC</td>
</tr>
<tr>
<td>E3,1</td>
<td>2001 June</td>
<td>Genesis RDBI binary code open to Si2/OAC</td>
</tr>
<tr>
<td>E4,2</td>
<td>2001 June 28</td>
<td>Genesis API specification released to the public as OA 1.0 API specs</td>
</tr>
<tr>
<td>E3,2</td>
<td>2001 June 28</td>
<td>Genesis RDBI binary code released to the public as OA 1.0</td>
</tr>
<tr>
<td>E2,6</td>
<td>2001 June 28</td>
<td>First downloads of the OA1.0 API specs and RDBI binary code from the public</td>
</tr>
<tr>
<td>E1,6</td>
<td>2002, Feb. 4</td>
<td>OAC released a roadmap and a timeline for making the OA RDBI source code public</td>
</tr>
<tr>
<td>E2,7</td>
<td>2002 Feb. 4</td>
<td>Cadence acquired veto power on OA API specs change decisions and released Genesis source code (OA 1.0) to the OAC.</td>
</tr>
<tr>
<td>E3,3</td>
<td>2002 Feb. 4</td>
<td>Genesis RDBI source code open to the public</td>
</tr>
<tr>
<td>E2,8</td>
<td>2002 Feb. 4</td>
<td>Synopsys expressed doubts about the quality of Genesis code and promoted the Milkiway RDBI from Avant.</td>
</tr>
<tr>
<td>E4,3</td>
<td>2002 Feb.</td>
<td>OA 2.0 API specs open to the Si2/OAC</td>
</tr>
<tr>
<td>E4,4</td>
<td>2002 March</td>
<td>OA 2.0 API specs released to the public</td>
</tr>
<tr>
<td>E3,4</td>
<td>2002 May</td>
<td>OA 2.0 RDBI source code open to the Si2/OAC (OA 2.0 RDBI is the cleaned and completely re-written in C++ Genesis code)</td>
</tr>
<tr>
<td>E2,9</td>
<td>2002 June</td>
<td>Synopsys completed acquisition of Avant.</td>
</tr>
<tr>
<td>E2,10</td>
<td>2002 July 31</td>
<td>Mentor filed patent infringement suit against Cadence and quit the OAC.</td>
</tr>
<tr>
<td>E2,11</td>
<td>2002 Nov. 14</td>
<td>Cadence and Synopsys settled trade-secret suit (about Avant)</td>
</tr>
<tr>
<td>E1,7</td>
<td>2002, Dec.</td>
<td>Golden Gate Working Group was built to bridge between OpenAccess and the Synopsys' Milkyway environment</td>
</tr>
<tr>
<td>E2,12</td>
<td>2002 Dec.</td>
<td>Synopsys joined Golden Gate Working Group to build a bridge between the OA and the open Synopsys-Milkyway environments</td>
</tr>
<tr>
<td>E3,5</td>
<td>2003 Jan.</td>
<td>OA 2.0 RDBI source code released to the public</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Event</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E4,5</td>
<td>2003 May 12</td>
<td>OA 2.1 API specs open to the Si2/OAC</td>
</tr>
<tr>
<td>E3,6</td>
<td>2003 May 12</td>
<td>OA 2.1 RDBI source code open to the Si2/OAC (enables production)</td>
</tr>
<tr>
<td>E4,6</td>
<td>2003 July</td>
<td>OA 2.1 API specs released to the public</td>
</tr>
<tr>
<td>E3,7</td>
<td>2003 July</td>
<td>OA 2.1 RDBI source code released to the public</td>
</tr>
<tr>
<td>E2,13</td>
<td>2003 Sep. 16</td>
<td>Cadence's Virtuoso design tool began to support both OA &amp; CDBA</td>
</tr>
<tr>
<td>E2,14</td>
<td>2003 Sep. 23</td>
<td>Mentor joins back the OpenAccess Coalition as part of the settlement of lawsuits with Cadence</td>
</tr>
<tr>
<td>E2,15</td>
<td>2003 Nov.</td>
<td>The first design tape-out using an in-house OA based tool kit</td>
</tr>
<tr>
<td>E2,16</td>
<td>2004 May</td>
<td>Mentor's Calibre began to support OA</td>
</tr>
<tr>
<td>E1,9</td>
<td>2004, July</td>
<td>Change decision veto power handed over from Cadence to the OAC</td>
</tr>
<tr>
<td>E2,17</td>
<td>2004 July</td>
<td>Cadence handed the veto power to the OAC change team</td>
</tr>
<tr>
<td>E4,7</td>
<td>2004 Nov.</td>
<td>OA 2.2 API specs open to the Si2/OAC members only</td>
</tr>
<tr>
<td>E3,8</td>
<td>2004 Nov.</td>
<td>OA 2.2 RDBI source code released to Si2/OAC</td>
</tr>
<tr>
<td>E2,18</td>
<td>2004 Nov.</td>
<td>Startup SiNavigator created an OA new business model</td>
</tr>
<tr>
<td>E2,19</td>
<td>2005 June</td>
<td>Synopsys joined OAC at director level &amp; began promoting OA</td>
</tr>
<tr>
<td>E4,8</td>
<td>2005 July</td>
<td>OA 2.2 API specs open to the public</td>
</tr>
<tr>
<td>E3,9</td>
<td>2004 Nov.</td>
<td>OA 2.2 RDBI source code released to the public</td>
</tr>
<tr>
<td>E2,20</td>
<td>2006 Sep.</td>
<td>Last version of Cadence's Virtuoso supports OA only</td>
</tr>
</tbody>
</table>

A visual representation of the development of the above events together with the links between them is shown in Figure 4-10.
4.6.2 Linkages between variables across the four layers

Relationship between OA API specification and RDBI code releases

Figure 4-11 illustrates the temporal development of the OA API specifications (Layer 4) and RDBI code (Layer 3) by showing together all incremental releases leading to the evolution of a specific major release.
Three interesting features of the OA standard development can be identified.

- At the initial stage of the OA standard evolution (2000-2003) it was the API specs that were released first followed by almost one year delay by the RDBI source code release. The first quarter of 2003 marks a new period in the development of the standard where the OA API specs started to be released together with source code.

- Examination of the OA 2.0 and 2.2 API specs and RDBI releases shows that in both cases at a given moment in time the API specifications stop evolving. From this moment on it is only the RDBI code that is being further developed without requiring OA API changes and mainly focusing on fixing bugs instead of adding new features. This is an indication of the maturity level of the specific API version.

**Relationship between numbers of downloads, membership and OA EDA tools**

Figure 4-12 illustrates the number of accumulated OA API and RDBI downloads, Si2 and OAC membership, and number of OA EDA design tools by year.

![Figure 4-12: Accumulated downloads, membership and OA tools by year](image-url)
Figure 4-12 suggests that the OAC membership, Si2 membership, the number of OA EDA tools, and the number of accumulated downloads including the OA API specs and the RDBI code are positively related. 2003 is the cutting point in time between the two periods in the development of the OA standard marking the significant increase in the Si2 membership growth and the availability of the first OA EDA tools.
5 DISCUSSION OF RESULTS

This chapter is organized into two sections. Section 5.1 provides answers to the three research questions. Section 5.2 discusses the insights gained while carrying out this research.

5.1 Answers to research questions

The first objective of this research was to answer the following three questions:

• What are the multidimensional constructs that could be used to describe the development of the OpenAccess standard?

• What are the differences between the characteristics of the OA reference database implementation and the characteristics common to open source assets?

• What is the relationship between the constructs used to describe the OpenAccess standard and the research literature?

Sections 5.1.1 to 5.1.3 provide answers to each of these three research questions.

5.1.1 What multidimensional constructs can be identified by examining the development of the OpenAccess standard?

Table 5-1 provides the five constructs drawn from examining the information in Chapter 4. For each construct, Table 5-1 shows the dimensions that define it and, for each dimension, the attributes documented in the case study.

The five constructs which were identified include: (i) Common asset, (ii) Project life cycle, (iii) Community, (iv) Company, and (v) Success definition.
Table 5-1: Constructs drawn from the Open Access project

<table>
<thead>
<tr>
<th>Construct</th>
<th>Dimension</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Common asset    | Pedigree    | Cadence first released its Genesis API as the 1.0 version of the OA standard (information model, API interface, and API specification) and the binaries of its reference database implementation to satisfy a request issued by the OAC in 1999 and then eight months later released the source code.  
  
The OpenAccess project subsequently released alpha, beta, production, and mature versions of the OA standard specifications as well as the binary and source code of the reference database implementation. |
| Rights          |             | Rights over which assets are accessible to whom, and what can be done with those assets are linked to membership type. Non members can download general releases of the OA standard specification and the source code of the reference database implementation. Academic members have access to member releases.  
  
  Si2 members can also use, reproduce and prepare derivative works of the OA standard specification and the reference database in both source and binary code for non-commercial and internal use.  
  
  In addition to what Si2 members can do, OAC members can reproduce, distribute and sublicense the OA standard specification and the reference database in both source and binary code for commercial use. OAC members can only include unmodified binary code in the products they sell. |
| Project life     | Phase       | Phase 1 starts with the founding of the OAC and ends when the OpenAccess project released to the public the Genesis API specifications as OA API specs 1.0 and the Genesis binary code as RDBI binary in June 2001.  
  
  Phase 2 ends when Cadence introduced the first commercial EDA design tool that supported the OpenAccess standard as well as other standards.  
  
  Phase 3 ends when Cadence introduced the first commercial EDA design tool to solely support the OpenAccess standard. |
<p>| Community        | Goals       | Established three goals that remained unchanged over the duration of the OpenAccess project. |
| Member structure |             | In 1999, two EDA companies (Cadence and Mentor Graphics) and six user companies were all part of one membership tier. In 2004, the structure was changed to one comprised of three membership tiers each of which had different degrees of responsibility and privileges. The purpose of the change was to increase the adoption of the standard. |
| Veto power       |             | Cadence received veto power over the change team as of February 4, 2002. This veto power was handed over to the OAC in July 1, 2004. |
| Contribution     |             | Si2 corporate members contributed most of the bug reports and new feature requests over the life of the project, contributions from the public were negligible. |
| User composition |             | Over time, the percentage of downloads from corporations and universities increased while the percentage of downloads from |</p>
<table>
<thead>
<tr>
<th>and growth</th>
<th>individuals decreased.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT support</td>
<td>Established and operated OpenEDA.org, a website that allowed contributors to communicate and contribute to the development of the OpenAccess standard.</td>
</tr>
<tr>
<td>Company Proprietary asset release</td>
<td>Motives for Cadence to release the OA standard specification and binaries of the reference data base included desire to: (i) influence the development of the OpenAccess standard, (ii) respond to customers' requests, (iii) increase the number of experienced developers working on asset development, and (iv) increase the likelihood that competitors will contribute to code development.</td>
</tr>
<tr>
<td>Risk</td>
<td>Candence exposed itself to when it released the OA standard specification and the reference database was that the company could no longer use the released assets as a source of differentiation.</td>
</tr>
<tr>
<td>Market offers</td>
<td>Companies developed and sold EDA tools for IC design that integrated the reference database or complied with the OA standard specification.</td>
</tr>
<tr>
<td>Internal use</td>
<td>Companies developed tools for internal and non-commercial purposes which integrated the reference database or complied to the OA standard specification.</td>
</tr>
<tr>
<td>Direct competition</td>
<td>Two competing standards were identified: Cadence's CDBA (in Phases 1 and 2) and Avant's Milkiway (in Phase 2).</td>
</tr>
<tr>
<td>Standard adoption</td>
<td>Motives for a company to adopt the OpenAccess standard included: (i) benefit from tight integration between different EDA tools (for end user companies) and high interoperability with complementary product (for EDA companies), (ii) respond to customers' requests, (iii) concentrate on core competence, (iv) quickly realize quality improvement at low cost, and (v) influence the development of the OpenAccess standard.</td>
</tr>
<tr>
<td>Risks</td>
<td>Risks of adopting the OpenAccess standard arose because the OA standard specification and the reference database (i) lacked complementary products, (ii) had limited functionality and (iii) had poor documentation.</td>
</tr>
<tr>
<td>Success definition Standard acceptance</td>
<td>Gain acceptance of the OpenAccess standard within the EDA industry and university research programs.</td>
</tr>
<tr>
<td>Number of OAC members</td>
<td>At the end of 2006, 37 companies were members of the OAC.</td>
</tr>
</tbody>
</table>

**Common asset**

Two dimensions were observed for the common asset construct: (i) Pedigree and (ii) Rights.

The Pedigree dimension refers to the line of ancestors of the assets the OpenAccess project
released to the public. Figure 5-1 shows that the OAC issued a request for technology during the fourth quarter of 1999. Cadence responded by providing assets related to their proprietary database known as Genesis. These assets were approved by the OAC as meeting its request for technology. In June 2001, the OpenAccess project released to the public the Genesis API specification (graphical representation of the information model, the C++ binding of the API interface, the HTML of the API specification) as OA API spec 1.0 and the Genesis binary code as OA RDBI binary.

Eight months later the OpenAccess project released to the public the source code of Cadence’s Genesis database.

The Rights dimension refers to the rights given to users of the common assets. Figure 5-2 shows the rights associated with each membership status. Non-members can download general releases of the OA standard specification and the source code of the reference database. A
general release refers to a version of an asset released to members 12 months earlier.

Academic members have access to member releases.

Figure 5-2: Rights to OA standard specs and RDBI by member status

<table>
<thead>
<tr>
<th>Membership Status</th>
<th>Non-Member</th>
<th>Academic</th>
<th>Si2</th>
<th>OAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Access (specs &amp; code)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General release</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Member release</td>
<td>(specs only, minimal fee)</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Production usage</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Redistribution of compiled code</td>
<td>As Downloaded</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Recompiled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Si2 members can also use, reproduce and prepare derivative works of the OA standard specification and the reference database in both source and binary code for non-commercial and internal use. In addition to what Si2 members can do, OAC members can reproduce, distribute and sublicense the OA standard specification and the reference database in both source and binary code for commercial use. OAC members can only include unmodified binary code in the products they sell.

Project life cycle

The second construct identified was named Project life cycle. Three phases were identified: (i) Origin, (ii) Competition, and (iii) Co-opetition (http://en.wikipedia.org/wiki/Coopetition). The "Origin" and "Competition" phases together form the first period (2000-2003) identified in Sections 4.3-5 and 4.6 characterized by slow membership growth and no development. The
"Co-opetition" phase coincides with second period (2003-2006) identified in Sections 4.3-5 and 4.6 characterized by fast membership growth and intensive OA community contributions.

Figure 5-3: Three phases of the OA project's life cycle

Phase 1: Origin (Q4, 1999 - June 28, 2001)

The founding of the OAC in the fourth quarter of 1999 marked the start of Phase 1 (Figure 5-4). Cadence and Mentor Graphics, the first and third largest EDA tool developers by revenue, were founding members of the OAC. Synopsis, the second largest EDA tool developer by revenue, did not join the OAC during Phase 1.

Shortly after the OAC was founded, it issued a request for technology searching for a proven API specification and RDBI that could be adopted as a standard. Cadence responded to the request for technology by offering to release to the public its Genesis RDBI. The OAC accepted the Cadence offer and Cadence assumed a leadership role in the development of the technology required to set the standard.
The OpenAccess project released to the public the Genesis API specs as OA API spec 1.0 and the Genesis binary code as OA RDBI binary in June 2001. Phase 1 ends with the release to the public of the first version of the OA standard.

Figure 5-4: The OA project’s life cycle Phase 1

During Phase 1, the attitude of other EDA tool suppliers was one of “wait and see”. The reason for this attitude was that the source code of the reference database was not released with the other assets.

Phase 2: Competition (June 28, 2001 - July, 2003)

In February 2002, the OAC agreed to provide Cadence veto power over the change team.
Mentor Graphics quit the OAC in 2002 due to a dispute with Cadence over a patent infringement.

Synopsys expressed doubts about the quality of the Genesis code and promoted the Milkiway RDBI developed by Avant as the standard. Avant was a company founded by former Cadence employees which had been acquired by Synopsys. Synopsys' acquisition of Avant, led to a lawsuit involving Cadence and Synopsys.

During Phase 2 (Figure 5-5), the two leading EDA suppliers were promoting two incompatible standards, Cadence’s Genesis and Avant’s Milkiway. A unified standard seemed an illusive goal. The Golden Gate workgroup was formed to build the bridge between the two standards.

The end of Phase 2 was marked by Cadence’s introduction of the first commercial tool that supported the Open Access standard, the new Virtuoso custom design platform. The tool had
previously only supported the Central Data Base Administrator (CDBA) standard. The Virtuoso technology had existed since Cadence was founded in 1987.


At the start of Phase 3 (Figure 5-6), various vendors introduced EDA tools based on the OpenAccess standard either for internal use or for commercial purposes.

In September 2003, Mentor Graphics rejoined the OAC. During Phase 3, EDA companies became the main contributors to the development of the OA standard. The number of members in Si2 and OAC as well as the number of commercial EDA tools introduced grew exponentially during this phase.

Cadence handed veto power over to the OAC change team in July 2004. This motivated other vendors to contribute to the development of new features for the reference database. A new
version of the OA standard known as 2.2 was released. It provided code with a higher level of maturity.

The end of Phase 3 was marked by Cadence introducing a new version of its Virtuoso custom design platform. The significant change was that this commercial release only supported one standard, the OpenAccess standard.

At the end of Phase 3, the top 5 EDA developers in the world - Cadence, Synopsis, Mentor Graphics, Magma and Zuken (Appendix B) - were members of the OAC.

**Community**

Community was identified as the third construct. It has six dimensions: (i) Goals, (ii) Membership structure, (iii) Veto power, (iv) Contribution records, (v) User composition and growth, and (vi) IT support.

The OpenAccess project had three clear goals which remained consistent over the life of the project. The goals included the provision of an IC design tool infrastructure, promotion of an open standard for IC design data access, and gaining adoption of the standard within the EDA industry and university research programs.

Membership structure was the second dimension of the Community construct. It refers to the member tiers that were established for the project. The project started with one member tier which included all EDA companies and end user companies. In 2002, the OAC membership was separated from the Si2 membership to work as an independent body to control the evolution of the OpenAccess standard. In 2004, the structure of the OAC was changed from a
one tier to a three-tier membership structure. The three membership tiers were defined as: adopter, partner, and director member. An adopter refers to a member who will use the OA standard specification and the reference database. The difference between partners and directors was voting rights. Directors could vote, while partners could not.

The change in membership structure was carried out for the purpose of attracting a larger number of EDA companies into the OAC. Each tier of the 3-tier membership structure addresses a different motive for a company to join the coalition and entailed different fees and degrees of responsibility and privileges. As a result of this change in membership structure, the membership of EDA startups increased from two to nine in 2004.

Veto power was the third dimension of the Community construct. Cadence received veto power over the change team as of February 4, 2002. This veto power was handed over to the OAC change team in July 1, 2004.

The fourth dimension of the Community construct was Contribution record. Table 5-2 shows that (i) Si2 corporate members contributed most of the bug reports and new feature requests over the life of the project and (ii) contributions from the public were negligible.

<table>
<thead>
<tr>
<th>Year</th>
<th>Contribution records</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Si2 corporate member</td>
</tr>
<tr>
<td>2004</td>
<td>182</td>
</tr>
<tr>
<td>2005</td>
<td>147</td>
</tr>
<tr>
<td>2006</td>
<td>163</td>
</tr>
<tr>
<td>Total</td>
<td>490</td>
</tr>
</tbody>
</table>
User composition and growth was the fifth dimension of the Community construct. Figure 5-7 shows that the downloads for the OA standard (OA standard specification and RDBI) from corporations and academic institutions have increased over time. In 2001, they accounted for 40% of total downloads while in 2006 they comprised 80%.

![Figure 5-7: Number of OA standard downloads](image)

Finally, the sixth dimension of the Community construct was information technology (IT) support. The OpenEDA.org is the website for the OpenAccess project. It was established on June, 2000. OpenEDA provides a centralized location on the web where users can discuss problems, track debugging, and submit requests.

**Company**

The fourth construct, Company, was comprised of five dimensions: (i) Proprietary asset release, (ii) Market offers, (iii) Internal use, (iv) Direct competition and (v) Standard adoption.

Proprietary asset release is the first dimension of the Company construct. It includes the motivations companies have to release proprietary assets to the public and the risks to which they become exposed when they do. Cadence released the OA standard specification...
(information model, API interface, and API specification) and binaries of the reference database in June 2001. The intent was to:

- influence the development of the OpenAccess standard
- respond to customers’ requests
- increase the number of experienced developers working on asset development
- increase the likelihood that competitors will contribute to code development

When Cadence released the proprietary assets it exposed itself to the risk that it could no longer use these assets as a source of differentiation.

Market offers was the third dimension of the Company construct. Market offers include commercial EDA tools that comply with the OA standard specification. Mature EDA companies and startups introduced 30 EDA tools that were compliant to the OA standard during Phase 3 of the OpenAccess project life cycle.

Companies also developed tools for internal and non-commercial purposes. These tools integrated the reference database or complied to the OA standard specification.

Direct competition was the fourth dimension of the Company construct. The construct referred to potential competitors to the OpenAccess standard in terms of companies and other standards. The largest EDA vendors include five companies. Two of these companies founded the OAC in 1999. Cadence joined in 1999. Mentor Graphics first joined in 1999, quit in 2002, and re-joined in 2003. Magma, Synopsis and Zuken joined in 2005. By 2005, all five companies were members of the OAC. Two competing standards were identified: Cadence’s CDBA (in Phase 1 and 2) and Avant’s Milkiway (in Phase 2).
Standard adoption was the fifth dimension of the Company construct. It included the motivations for companies adopting the OpenAccess standard and the risks to which they were exposed when they do. Motives for a company to adopt the Open Access standard included:

- benefit from tight integration between different EDA tools (for end user companies) and high interoperability with complementary product (for EDA companies)
- respond to customers' requests
- concentrate on core competence
- quickly realize quality improvement at low cost
- influence the development of the standard by the open source community

Risks of adopting the OpenAccess standard arose because the OA standard specification and the reference database:

- lacked complementary products
- had limited functionality
- had poor documentation

Success definition

The fifth and last construct was Success definition. This construct was comprised of three dimensions: (i) Standard acceptance, (ii) Number of OAC members, and (iii) Number of top EDA vendors actively participating in the development of the Open Access standard.

Standard acceptance was one of three objectives of the OAC. The OAC sought to gain acceptance of the OpenAccess standard within the EDA industry and university research programs. If the Open Access standard becomes the only standard supported by all
commercially available EDA design tools, this will be a clear indication of its complete adoption.

Number of OAC members was the second dimension of the Success definition construct. At the end of 2006, 37 companies were members of the OAC. Top EDA company participation was the third dimension of the Success definition construct. In 2005, the five top EDA companies had joined OAC. Cadence joined in 1999. Mentor Graphics first joined in 1999, quit in 2002, and re-joined in 2003. Magma, Synopsis and Zuken joined in 2005.

5.1.2 What are the differences between the characteristics of the OA reference database implementation and the characteristics common to open source assets?

The researcher concluded that the reference database implementation supporting the OpenAccess standard is not an open source asset. To reach this conclusion, the researcher examined whether or not the distribution terms of the reference database implementation code complied with the criteria for the definition of open source provided by the Open Source Initiative. The criteria are published at http://www.opensource.org/docs/osd. The analysis was performed for OAC members, Non-members, Academic members and Si2 members.

Table 5-3 provides the researcher's assessment of whether or not the distribution terms of the reference database meet the criteria at http://www.opensource.org/docs/osd
<table>
<thead>
<tr>
<th>Definition</th>
<th>Non-member</th>
<th>Academic member</th>
<th>Si2 member</th>
<th>OAC member</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Free redistribution - selling or giving away the software as a</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>√</td>
<td>It is only OAC members who could distribute the OA RDBI code in binary form only. Non-members, academic members and Si2 members who are not members of the OAC cannot distribute anything.</td>
</tr>
<tr>
<td>aggregate software distribution containing programs from several different sources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Source code - the program must include source code, and must allow</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>√</td>
<td>It is only OAC members who could distribute the binary code. There is, however, a well-publicized means for anyone interested who signs the OA license to obtain the source code by downloading it via the Internet.</td>
</tr>
<tr>
<td>distribution in source code as well as compiled form.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Derived works - the license must allow modifications and derived</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>OA community members can modify the RDBI for internal and non-commercial use only. No one is allowed to distribute modifications of the current official RDBI release.</td>
</tr>
<tr>
<td>works, and must allow them to be distributed under the same terms as the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>license of the original software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Integrity of the author’s source code - the license must explicitly</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>No one is allowed to distribute the OA RDBI source code.</td>
</tr>
<tr>
<td>permit distribution of software built from modified source code.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. No discrimination against persons or groups</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>This open source criterion does not apply since there are different rights for different types of members (levels of membership).</td>
</tr>
<tr>
<td>6. No discrimination against fields of endeavour - the license must</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>√</td>
<td>Non-members and academic members can use the code for non-commercial use only. The Si2 corporate members can use the code for internal use only. OAC members are the only ones who could commercially use the code.</td>
</tr>
<tr>
<td>not restrict anyone from making use of the program in a specific field</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>of endeavour - in a business, in genetic research or other.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Distribution of license - the rights attached to the program must</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>The OA software can be distributed as part of a product to customers in binary form only by OAC members only. An OAC member’s customer willing to distribute the binary code must become an OAC member.</td>
</tr>
<tr>
<td>apply to all to whom the program is redistributed without the need for</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>execution of an additional license by those parties.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. License must not be specific to a product - the rights attached to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>program must not depend on it being part of a particular software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>distribution.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. License must not restrict other software - the license must not</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>√</td>
<td>This term applies to OAC members only since it is only them who can distribute the OA RDBI binaries.</td>
</tr>
<tr>
<td>place restrictions on other software that is distributed along with the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>licensed software.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. License must be technology-neutral - no provision of the license may</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>The OA RDBI software must always comply with the OA API.</td>
</tr>
<tr>
<td>be predicated on any individual technology or style of interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.3 shows that the distribution terms of the reference database for OAC members do not meet criteria 3, 4, 5, 7 and 10. Criteria 3, 4 and 7 shown in Table 5.3 are not met because the distribution terms of the reference database for OAC members do not allow members to distribute derivative source code of the reference database. For commercial purposes, members are only allowed to distribute unmodified binary code. Criterion 5 shown in Table 5.3 is not met because the rights to use and distribute the reference database implementation are different for different membership levels. For example, companies willing to distribute the unmodified version of the OA RDBI binary code must become OAC members and pay a membership fee. Criterion 10 shown in Table 5.3 is not met because the source and binary code of the reference database must comply to the OA standard specification. Thus, this code must comply with the particular style of interface specified by the OA standard specification which is written in C++.

Table 5.3 also shows that the distribution terms of the OA reference database implementation for Non members, Academic members and Si2 members do not meet the criteria of the open source definition at http://www.opensource.org/docs/osd.

The comparison given above shows that the main differences between the characteristics of the OpenAccess reference implementation and the characteristics common to open source assets is due to the differences in the license terms associated with the distribution and the rights to use the open assets.
5.1.3 What is the relationship between the multidimensional constructs identified and the research literature?

Relating constructs to user innovation networks

von Hippel and von Krogh (2001) and von Hippel (2002) have identified five dimensions of user innovation networks. These are:

1. Free revealing of a technology or asset (i.e., release of an existing technology or asset to all interested agents without imposing a direct payment).

2. User innovation community (i.e., group of individuals or companies motivated to participate in the use and improvement of the released technology or asset driven by an expectation to derive some benefit, either from the innovation itself or from the process of innovating).

3. Collective invention process (i.e., a cyclic process of follow-on innovation leading to a series of freely revealed incremental improvements to the technology or asset triggering new rounds of innovation activities).

4. Commons-based peer production (i.e., a newly emerging mode of production in which the members of user community collaborate on projects leading to improved or completely new versions of the released technology or asset).

5. User community support (i.e., Governance and management mechanisms establishing and maintaining an effective community structure that maximizes other users’ motivation to contribute to further development).
The five dimensions identified by von Hippel and von Krogh (2001) and von Hippel (2002) highlight characteristics of the process a community uses to develop a common asset. Three of the five constructs identified in Table 5-1, Project life cycle, Community and Success definition, also highlight the characteristics of the development process. The other two constructs shown in Table 5-1 highlight the nature of the common asset produced (Common asset) and the interactions between a company and the (i) development project and (ii) market (Company).

The five dimensions identified by von Hippel and von Krogh (2001) and von Hippel (2002) are all present in the OA OpenEvolution process, however they are expressed at a much higher level of abstraction than the dimensions shown in Table 5-1.

**Relate the findings to Moore’s view of the technology life cycle**

Moore (2000) conceptualized the technology life cycle in terms of changes in the nature of the product being bought by customers. The conceptualization used to identify the three phases discussed in section 5.1.1 is similar to that used by Moore in that the end of a phase is marked by the introduction of product intrinsically different than the product that marked the start of the phase.

**Relate the findings to the measures of open source project success**

Open source literature defines success using measures such as number of downloads, number of contributions, and number of new releases (Crowston et al., 2003; Rehman,
To date, development project success has not been defined in terms of the need to establish the outcome of the project as a standard. Defining success as standard adoption has important implications for the way the development project is managed.

Relate the constructs to the diffusion of innovation

Rogers (1983, 1995, 2003) has argued that five attributes of an innovation affect innovation adoption. These attributes are: relative advantage, compatibility, complexity, observability and trialability.

It can be argued that all five of Roger's attributes were identified in the case presented in chapter 4. The OpenAccess standard provides relative advantage to both EDA companies and end-user companies. The standard eliminates the need to invest in proprietary API, data model and database solution, enhances interoperability with vendor tools, frees up resources to focus on developing tools that add value to customers, and leverages collective knowledge at a lower cost.

The OpenAccess standard was designed to be compatible with the infrastructures with which EDA and end-user companies were familiar. For large and small EDA companies, the OpenAccess standard made it easier to access their customers and reduced the effort required to interface with them. The standard also reduced integration costs.
The OpenAccess standard reduced complexity by providing single storage per data type. This eliminated duplication, conflict and confusion. The standard’s integrated database also made it easier to manage a large distributed team.

The development process used by the OpenAccess project increased the observability of the standard. The process used to develop the standard was visible. Contributions and the various releases of the assets to the public increased the observability of the standard.

Potential users could download the standard and try it out before deciding to adopt it. This increased the trialability of the standard.

**Relate the findings to the motivations to release proprietary assets as open source**

To set standard, build better relations with customers, nurture innovation, and trigger gift economy effect are identified motivations for companies to release code as open source (Goldman & Gabriel, 2005; Henkel, 2004; Koenig, 2004). Similar motivations were observed in the case of Cadence’s release of its Genesis database as the OA standard.

**Relate the findings to the motivations for companies to adopt open source**

To set standard, use open source software as a low cost component, and nurture innovation are identified motivations for companies to participate in open source
software project (Bonaccorsi & Rossi, 2004; Henkel, 2004). Similar motivations were mentioned by OAC member companies.

**Relate the characteristics of the OpenAccess standard to the definition of open standards**

The Open Access standard was designed as an open standard. Although its license terms do not formally meet the 10 criteria of the open source definition, its development practices qualify it as a “level 3” open standard (Tiemann, 2006). The development style of the OA reference implementation provided i) a means for advancing the standard over time as practices improve, and ii) a safeguard against the development (forking) of multiple API definitions.
5.2 Insights gained while carrying out the research

Ten insights were gained while carrying out this research. These insights are:

**Veto power over the change team accelerates technology development but slows down standard adoption**

Providing a company with veto power over the change team accelerates technology development that is consistent with the goals of the standard setting organization and the company that holds the veto power. However, other companies develop a “wait and see” attitude towards standard’s adoption. This slows down the adoption process.

Adoption of the standard accelerates when no company holds veto power over the change team.

**Requirements to adopt a standard prevent the reference implementation code released to the public from meeting the criteria for open source software**

The standard adoption process requires that the code of the reference asset be distributed with no modifications and that it is compliant to an interface implemented with a particular technology. These two requirements prevent the asset from being an open source asset. Thus, the requirement that for a software to become a reference implementation for a standard prevents it from meeting the criteria for the definition of open source provided by the Open Source Initiative.
Open development processes do not necessarily lead to the production of open source assets

The OpenAccess standard was developed using an open development process referred to as Open Evolution Process. This process did not lead to the development of open source assets.

Project success is defined in terms of standard adoption

The success of the OpenAccess project was defined more in terms of the adoption of the standard by companies in the target industry than by the number of downloads, number of developers, and number of releases of its API specs and reference implementation code.

The phases of the life cycle of standard development can be identified using product-introduction milestones

Three phases of the life cycle of the development of the OpenAccess standard were identified. The start and end of each base was marked by the introduction of a new product which was qualitatively different than the previous one. The end of Phase 1 was marked by the release of the OA standard specification and the code binaries of the reference database implementation. The end of Phase 2 was marked by the introduction of the first commercial tool that supported the OpenAccess standard as well as another standard. The end of Phase 3 was marked by the introduction of the first commercial tool to solely support the OpenAccess standard.
Examining solely the characteristics of the process used to develop a common asset may not be enough to understand user innovation networks. The researcher argues that to understand user innovation networks, we must understand the pedigree and rights of the common asset being produced, the interactions between companies and the (i) development project and (ii) market as well as the characteristics of the development project. Knowing only the characteristics of the open development process is not enough.

Duration from when a proprietary asset is released to the community to when it becomes an open asset is approximately 18 months

The transition from proprietary to open assets does not occur immediately, it takes months. The duration of the transition of the reference database from proprietary to open was approximately 18 months. The binary code for the Genesis database was released in June 28, 2001. The first release that did not include any of the proprietary code included in the original release was introduced in January 2003, 18 months later.

A multiple tier membership structure accelerates standard's adoption

The number of members in Si2 and OAC grew significantly when the structure was changed from a single tier to a three tier membership structure. Each membership structure targeted a particular type of member. This made it easier for companies to join a membership tier with which they felt most comfortable.
Standard adoption is driven by end-users at the early phases and then by tool vendors at the latter stages.

Figure 4-6 shows that the number of members considered to be users (Semiconductor and electronic system companies) dominated membership in 2000-2002, the early years of standard adoption. Mature EDA companies and EDA and startups dominated the latter stages of standard adoption. Moreover, Figure 4-7 shows that users introduced a greater proportion of tools in 2003 and 2004 than afterwards.

Pedigree includes multiple forms of the community assets

The pedigree of the components of the OpenAccess project shows that the assets released to the community had various forms. These forms included: (i) source code, (ii) binary code, (iii) graphical representations, and (iv) HTML pages.
6 CONCLUSIONS

6.1 Conclusions

This research answers three questions: i) What are the multidimensional constructs that could be used to describe the development of the OpenAccess standard?, ii) Is the reference database implementation that supports the OpenAccess standard an open source asset?, iii) What is the relationship between the constructs used to describe the OpenAccess standard and the research literature? To answer the questions, a multi-level, longitudinal study method was used to examine seven years of data about the OpenAccess project.

Five multilevel constructs and their constituent dimensions were identified and linked to the existing literature. The results suggest that:

- using open development processes contributed significantly to the development and adoption of the OpenAccess standard
- the open processes used for the development of the OpenAccess standard resulted in the production of assets that do not comply with the definition of open source
- the main difference between the characteristics of the OpenAccess reference implementation and the characteristics common to open source assets is due to the differences in the license terms associated with distribution and use
- the differences in the characteristics mentioned above were intentionally implemented to help the development and the integrity of the OA standard.
6.2 Limitations

The research has at least three limitations.

First, this study only examined the OA license version 4.0. This is due to the fact that previous OA licenses were not accessible. The discussion of the OA licenses terms was based on this version only. Examining previous license versions would have provided more insights into the temporal dynamics of the OA standard development.

Second, the data in this research was collected from secondary sources. Companies were not directly contacted and executive managers were not interviewed. Data from interviews would have provided additional insights into the motives and risks of different types of companies to participate in the development and adopt the standard.

6.3 Future research

The following suggestions are made for future research:

- Using industry relevant variables such as revenue and profitability of different company types (EDA, electronic systems and semiconductor) to examine the effect of standard adoption.
- Using data on the evolution of the license terms and relating it to the standard development and adoption processes.
REFERENCES


APPENDICES

Appendix A: The OpenAccess project case study

The subject of this case study is the OpenAccess project of the Silicon Integration Initiative (Si2) which is driving the adoption of the OpenAccess API design data standard. The Si2 is a user-driven organization involving companies from the semiconductor, electronic systems, and electronic design automation (EDA) industries. Its main goal is to address advanced interoperability and integration issues across silicon design flows within and between these industries. Si2 member companies work together to identify common issues and sponsor projects to reduce the cost and complexity of silicon design and manufacture of semiconductor integrated circuits. The development and the promotion of the adoption of the OpenAccess API standard is the first example of these collaborative projects.

1. Introduction

The evolution of the semiconductor, electronic systems, and EDA industries is associated with an increasing complexity of the integrated circuit design and manufacturing processes. Due to the shrinking feature sizes, it becomes more and more difficult for a single EDA vendor to provide a complete design solution to meet the highly demanding requirements of its customers. By the late 80’s, it was noted that the cost for semiconductor companies to integrate an EDA application into the in-house environment was often over twice that of the application itself. Today, the necessity the time for the translation of design files between design tools from different vendors is approximately estimated as 5 times the tool cost (Goyal, 2006). What is even more critical is that the time wasting on the format conversion is always associated with a larger price reduction from being late to market. The complicated design rules, rapid increasing file sizes, and the sequential nature of the design process challenged the translator-based design flow to continually serve the industry in a cost-effective way. To
totally solve these bottlenecks, the industry initiated a variety of efforts to standardize the design data format, including its Application Programming Interface (API) and information model.

The first attempt to provide an industry-accepted standard and technology enabling interoperability of EDA applications and design data for end-users and suppliers worldwide was made by the creation of the CAD Framework Initiative (CFI) in 1988. In 1993, the CFI published its design data representation (DR) API standard. The activities associated with the promotion of the DR API standard were mainly grounded within the CFI committee and, thus, limited by the CFI resources. The DR provided an information model for logical level design data only. Without the standardization model of physical data, electrical data, and manufacturing records, it did not achieve widespread adoption across the commercial EDA industry.

In the mid-1990's there was a second attempt to provide an industry standard - the Technical Data of Chip Hierarchical Design System (CHDStd) - sponsored by SEMATECH (SEmiconductor MAnufacturing TECHnology), a non-profit consortium in the form of a partnership between the United States government and 14 US-based semiconductor manufacturers to solve common manufacturing problems and regain competitiveness for the US semiconductor industry. The main goal was the revival of the CFI DR API standard but SEMATECH decided to solicit a working solution from industry rather than developing a new one. After consideration, a technology from IBM (called the Integrated Design Model – IDM) was chosen as a basis. The IDM technology not only successfully provided rich data scope but also was efficiently shaped to meet all of the CHDStd requirements. Unfortunately, in a way similar to that of the CFI DR API case, the specification failed to gain industry acceptance.
The main reason for that was identified simply as an issue of return-on-investment (OA book, 2006). Because CHDStd was just a paper specification and there was no compliant reference database software code to support its implementation, the cost to refit commercial EDA applications to CHDStd was perceived as being too high.

At the 34th Design Automation Conference (DAC, 1997) the CFI changed its name to Silicon Integration Initiative (Si2) and repositioned itself as a user-driven consortium of industry and academic members to address advanced interoperability and integration across silicon design flows. The mission of the Si2 was set to “achieve industry adoption of collaborative technology and services that deliver higher levels of silicon design integration enabling compelling advantages for its members through reduced costs, faster time to market, and improved IC design capability.” In late 1999, to find a way to make CHDStd successful, SEMATECH asked the Si2 to take ownership of the CHDStd program and Si2 accepted this challenge and created the OA coalition as part of its organizational structure to manage the newly created project called OpenAccess. The OA coalition was designed to operate under Si2 bylaws.

2. The OpenAccess project

2.1 The OpenAccess concept

The Si2 positioned the OA project as a community effort to provide true interoperability, not just data exchange, among IC design tools through the development and the adoption of an open standard design data API and reference database supporting that API. Three main goals were set as follows:
• provide an IC design tool infrastructure that yields integrated systems rather than sequential flows, choice of design tools and provider, technology transfer of innovative research, and collaborative design capability for ICs
• promote an open standard for IC design data access
• gain adoption of the standard within the EDA industry and university research programs.

EDA, electronic systems and semiconductor industry players were invited to actively participate in the OA project becoming formal members the OA Coalition (OAC). The Si2 identified the proactive development of specific adoption mechanisms as critical for the success of the OA standard adoption. Both, the CFI standard and CHDStd, technically provided an interoperable framework enabling the construction of high efficiency interoperable flows, but neither of them could prevail in the commercial industry. Compared with the CFI DR API standard and the CHDStd, for the first time, the Si2 made a significant addition to the goals of the OA project - to “eliminate a key barrier to adoption within the EDA industry and university research community via ready availability of a commercially viable implementation of the API” (OA book, 2006, p. 4) To address this and other important issues affecting adoption – critical to the success of any industry standard – the OpenAccess founders established the following business requirements:

• make the API specification publicly available to anyone at low or no cost
• provide an unbiased change management procedure for the API
• submit the OA specification to an accredited standards body for publication as an industry standard
• assure that use of the API enables customer choice of any compliant database and does not require adoption of any particular database
• provide access to a high-quality reference database implementation, including
o binary code – for interoperability verification as well as reducing deployment and ramp-up costs
o source code – freeing development and enhancement opportunities to users

2.2 Three building blocks of the OpenAccess standard

To achieve a high level of adoption, the Si2 crafted the OA standard by means of three primary building blocks (Figure 1).

**Figure 1: Primary building blocks of OA project**

Source: OpenAccess™ Release 2.2 Standard API Tutorial, 4th Edition (OA 2.2.6) (with permission of the Si2)

- a) an openly documented API Standard

It should be pointed out that the API specifications represent the main content of the OpenAccess Standard which includes three components defining the interface:

- an information model defined by a collection of entity relationship diagrams and describing a conceptual perspective of the objects and their mutual relationship including the navigability across object relationships
- a data model defined by C++ header files, which specify software class and function interface details
- the API specifications which presents the header information in a more
readable format and includes additional constraints.

This specification is automatically generated from the actual C++ headers. Information such as Class inheritance trees, functions and their arguments, specially formatted constraints are auto-extracted into a convenient HTML documentation that is extensively hyperlinked. This technique ensures centralization of the specification details directly tied to the actual implementation code.

b) a reference database implementation of the standard

The OA reference database implementation (RDBI) is the source code for, or linked libraries of, the implementation of the OpenAccess API that is publicly available as part of the OA distribution. It is possible that more than one RDBI of the OA API could be written, perhaps by different vendors. The OA RDBI was designed to be developed as just one particular implementation of the API, with publicly available source, updated in sync with any changes to the API standard. It was expected that the OA RDBI will be an enormous convenience that can be used as is or as a vehicle to aid in the development and test of other databases complying with the OA API standard. According to the Si2, the intent of the OA RDBI was to help OpenAccess adopters become more successful by:

- transferring technology to the public that encompasses high-end, commercially successful design database and product experience
- demonstrating the strong future commitment to OA from a major EDA vendor
- offering an extraordinarily low-cost quick-start option for: i) experienced companies with internal databases that might be viable but due to their age could benefit from a next-generation implementation; ii) newer or smaller companies looking to leverage the significant engineer-year resources (to develop their own database) that the reference implementation encapsulates.
It should be pointed out that the OA reference implementation is not a formal part of the OA API standard and was designed just as an adoption and standard development mechanism in accordance with the initial goals of the Si2/OAC and in contrast with the two previous unsuccessful attempts - the CFI standard and CHDStd. What is really standardized here is the API specification and not its reference database implementation – i.e., in this sense, the situation is the same as in the CFI standard and CHDStd.

c) a detailed OpenEvolution process to manage on-going development, distribution, and support of both the API and the reference implementation

A carefully managed governance process was designed as a key to the commercial adoption of the OpenAccess API standard including:

- centralizing the information related to the API Standard, Reference Implementation, and auxiliary components, including bug tracking, enhancement requests, information exchange, and educational materials
- coordinating the development of specifications and code as they evolve to cope with the needs of the industry, including support of working group and other collaborative activities
- distributing the software code, documentation, and related information via web-based technologies
- protecting the rights of participants in the OpenAccess Coalition through appropriate joint use agreements.

To address these needs, the OpenEvolution process (Figure 3) was built based on a two-layer governance structure as the collaborative engine driving the OpenAccess project.
At the heart of OA project governance structure are several groups of stakeholders controlling the evolution of the API standard in terms of funding as well as technical guidance including:

- the OAC members\(^6\) - a body composed by any company or bona fide industry organization who commits in contributing engineering and financial resources to the development, evaluation, and adoption of OpenAccess API (each member has one vote right on any decision of project activities and related processes)

- the Change Team members – a body of technical experts who serve the OAC to manage the evolution of OpenAccess API (each member of the Change Team is periodically elected from a OAC company and operates under the specific rules spelled out in the Coalition and Change Team bylaws)

- the Integrator - Cadence serve as the integrator of all OpenAccess RDBI. (once a set of changes has been approved by the Change Team, Cadence has the first right of refusal to provide a just Cost/Time estimate within 15 business days for

---

\(^6\) Appendix A provides a detail description of OpenAccess community governance structure.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
integrating the changes to the RDBI; in case of refusal, the OAC has the option to having the changes integrated through an OAC member third party company)

- the Working Groups – bodies enabling the joint cooperation of many different professionals with expertise spanning many specialties to anchor around the specified task (analysis, deliberation, and recommendations regarding enhancement requests are delegated to working groups convened by the Change Team).

2.3 Open source principles of the RDBI

Surrounding the core of active participants involved in the OAC management, there is the OpenAccess Community including anyone who wishes to use or contribute to OA under the terms of the OA license (Table 1). A broad OA community base of talent and resources (with a carefully designed layered membership structure shown in Table 8 and Figure 4 provided the necessary mechanisms for downloading the OA API standard specification and RDBI, logging bug reports and enhancement requests, or simply asking questions in a familiar open forum environment. An open source community model was developed by sequentially enabling the main characteristics of open source software development (i.e., open access and downloading, use, modification and redistribution of code under the specific terms of the OA license) to benefit the RDBI from a potentially large pool of support for on-going enhancements and fixes.

The Si2 developed a two-layer structure of the OA project comprising the OA API specifications and the OA RDBI by following two firm principles: i) all changes of the API must be suggested through an implementation in the current RDBI, and ii) the OA RDBI must always comply to the OA API standard. These two principles together with the benefits
enabled by the open source development characteristics of the RDBI were intended to turn the RDBI into an efficient standard development and adoption mechanism. The strict OAC’s management process ensured stability of the implementation code and continuous compatibility of the OA standard.

Table 1: OA license terms

<table>
<thead>
<tr>
<th>Si2/OAC membership</th>
<th>Non exclusive, royalty free, worldwide, copyright license to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Si2 member</td>
<td>Use, reproduce and prepare derivative works of the software contained in the Package and the API specification for such software in both source and binary code forms solely for Non-commercial Use</td>
</tr>
<tr>
<td>Si2 academic member</td>
<td>Use, reproduce and prepare derivative works of the software contained in the Package and the API specification for such software in both source and binary code forms solely for Non-commercial Use within the scope of member’s academic activities</td>
</tr>
<tr>
<td>Si2 corporate member</td>
<td>Use, reproduce and prepare derivative works of the software contained in the Package and the API specification for such software in both source and binary code forms solely for Internal Use</td>
</tr>
<tr>
<td>OAC member</td>
<td>Reproduce, distribute and sublicense the unmodified software contained in the Package in binary code form only and the unmodified documentation contained in the Package, and to include in any such distribution bona fide error corrections to the software in binary code form only and documentation contained in the Package that do not alter the syntax, semantics and/or behavior of the API specification for such software, solely in connection with the offerings that Recipient markets as part of its product and services offerings in the normal course of its business, which licenses shall remain in effect in perpetuity unless terminated in accordance with this Agreement.</td>
</tr>
</tbody>
</table>

Source: OpenAccess – Internal Use and Distribution License V 4.0 (with permission of the Si2)

Figure 3: OA licensing rights: Access/Usage/Redistribution
Technically, an API change is implemented as follows. If the user makes enhancements to the RDBI and wishes them to be part of the OA API standard package, then the changes are submitted back to the Change Team. The Change Team considers all such requests and, if accepted, forwards them to the OpenAccess Integrator for incorporation into the standard OA package. Accepted changes may be incorporated as submitted by the user or modified by the Integrator for consistency with the OA architecture. A working group may be convened for further study to ensure sufficient generality in the final solution. If a new feature is requested without accompanying implementation code, the OAC may consider funding or a request for technology (RFT) to assist in the development. Approved enhancements are published in a preliminary developed Roadmap along with an approximate schedule. Once a contribution is adopted or a change request is implemented, it is incorporated into a standard OA release and published on the official OA website. Even if a requested enhancement is ultimately rejected during the Change Team review process the Open-Access extensibility features can usually be used to accomplish the desired result for the user, without resorting to a change in the OA core.

2. Chronological description of events

OpenAccess founded

The Si2 created OpenAccess project in the 4th quarter of 1999 as the third attempt to build an EDA, electronic systems, and semiconductor industry wide standard. The OpenAccess Coalition (OAC) was initially formed as the Design API Coalition (DAPIC) to lead this project by operating under Si2 bylaws. Its 8 founding members included the following EDA and semiconductor industry leaders: Cadence, Mentor Graphics, HP, IBM, Intel, Lucent, Motorola and Sun. Cadence and Mentor Graphics are the first and third largest supplier of EDA tools (by revenue 2005).
Open EDA website launched

After a lively discussion on open-source approaches to standard development at the Design Automation Conference (DAC) on June 7, 2000, during the open source EDA DAC conference panel session the Si2 officially announced the launching of launched an open source web site for the EDA and user communities called OpenEDA, which provided access to standards and documentation under an open source license. The initial offering on the OpenEDA.org website site included the releases of Library Exchange Format (LEF) and Design Exchange Format (DEF), 2 physical design formats from Cadence Design Systems.

The Si2 expected the OpenEDA Open Source Web Site to not only allow a true EDA open community site that offers companies, universities, and others the ability to post open source code, track debugging, and promote EDA interoperability, but also to provide a way to move interoperability efforts forward from a committee level to a community level.

Request for technology and agreement with Cadence

As soon as OpenAccess was founded, the OAC immediately issued a request-for-technology, looking for a proven application programming interface (API) specification and reference database implementation that could be commercially adopted supporting a broad set of applications and released as open source. As a result of a collaboration with the OpenAccess Community, Cadence responded, offering its API specification for its Genesis RDBI to the OAC in September 2000. OAC accepted Cadence’s response to the RFT providing and in June 2001 finally entered into a community-source agreement to release its next-generation Genesis reference database implementation source code as the basis for the OA standard. The
Genesis API specification and reference database source code were renamed to “OpenAccess API” and “OpenAccess reference database,” respectively.

**The release of G1 API standard**

After OAC received the contribution of the binary code of Genesis RDBI from Cadence, the version 1 of the OpenAccess API specification and reference database implementation in binary form only were published for public review and download on the OpenEDA.org Web site in June 28, 2001. The OA standard version 1 was called G1 (Gensisl) because it was the original Cadence Genesis API specification and binary reference database (no source code) implementation.

**OA source code release roadmap**

At the beginning of February 2002, the OpenAccess Coalition released a long-awaited roadmap for the source-code availability of Cadence’s Genesis database. According to the published OpenAccess project timetable, the full, production version of the OA RDBI was not to be available until December 2003, and Cadence retained veto power over changes until July 2004.

The initial responses from the industry were mixed. For the OpenAccess advocates - the most powerful group of EDA consumers such as Intel - the timetable meant that the effort is not theoretical anymore and both users and EDA vendors could start planning. EDA companies such as Simplex Solutions Inc., Nassda Corp., Synchronicity Inc., Neolinear Inc. and Synplicity Inc. expressed their support for the OAC. Mentor Graphics Corp., though a charter member of OpenAccess, declined to comment. Magma Design Automation Inc. and Monterey Design Systems Inc. were both taking a wait-and-see approach.
The release of G1 database source code

On February 4, 2002, Cadence contributed G1 (OpenAccess version 1) source code along with a refreshed set of matching binaries to the OpenAccess Coalition. G1 source code was the original Cadence Genesis code, and it was for digital designs only. It was intended primarily for proof-of-concept purposes, training and for gaining familiarity with data models. It was not supposed to be upwardly compatible with the next OpenAccess version 2.

With the database source code in place, the focus shifted to make the OA standard real. Until February 2002 the EDA vendors that publicly joined OpenAccess were Mentor Graphics Corp. and Simplex Solutions Inc. Some of the other EDA vendors continued to voice concerns. Noting that other tools such as Synopsys tools still use closed databases. Bob Smith, vice president of marketing at Magma Design, said Magma will “wait and see” how OpenAccess evolves before deciding to jump in. Cadence’s main competitor Synopsys expressed strong reservations: Rich Goldman, Synopsys’ Vice President of strategic market development, termed Cadence’s ongoing control over the OA standard “troubling” and expressed his concern for existing unresolved questions about the quality of the Genesis code. 18 months ago, at the Design Automation Conference (June, 2000), Synopsys had been open to joining OpenAccess. But now Synopsys announced that they would acquire Avanti and would not sign on to the OpenAccess community without the availability of Genesis source code. The main message was that they would rather use Avanti’s long-established Milkyway database and work towards the integration of the two companies’ offerings.

In June 2000 Avanti had offered to license the Milkyway API to third-party EDA vendors, but had not committed to release source code or provide open-source access. As the major
competitor of Genesis, Milkyway did not meet the OpenAccess criteria for openness - it needed to i) be outside the control of the donating party; ii) be available to anyone without discrimination; and iii) be offered under “reasonable” terms and costs. Cadence had already pledged that the Genesis RDBI will comply with all the above requirements.

The Synopsys-Avanti merger appeared as a danger for the emergence of two competing standards and two back-end environments out there. It posed a challenge for the OpenAccess community because Milkyway would take momentum from version 1 of the OA API standard and RDBI, which was not yet commercially deployed at that time. It would enable EDA users to choose between two incompatible back-end databases – OpenAccess (Genesis) and Milkyway.

**The release of OA 2.0**

In February 2002, Cadence released the OpenAccess version 2 (OA 2.0) API specification to the OpenAccess Coalition and, following to the initially announced Roadmap, the document was available on OpenEDA.org one month later. In May Cadence also released version 2 of the OpenAccess database source code. After some minor adjustments and additions, Cadence delivered to the OAC the OA 2.0 RDBI source code including the infrastructure needed to create database programs in December 2002. It is this December release that provided the coalition with what it needed to go into “production release” mode. Through the evaluation and validation of beta testing the coalition finally released the OA RDBI including source code and binaries to the public in January 2003.

Beginning form version 2 of the OA RDBI code, there was no Cadence proprietary code any more. Beyond the G1 (working for digital designs only) the OA 2.0 standard supported both
analog and mixed-signal designs. In addition, this release also provided the necessary infrastructure to meet the technological requirements for leading IC design supporting new levels of capacity and performance, 64-bit thread-safe C++ architecture and powerful extensibility. Although the OA 2.0 standard was not a final, production release, it allowed the users to start building EDA applications based on the existing OA RDBI.

Until 2002, OpenAccess Coalition got the support from the most powerful group of EDA consumers it had ever had. Many user companies such as Sun Microsystems, Philips Semiconductors, Tektronix Inc., and Mitsubishi joined the OAC since its founding. But the user-backed OAC still lacked broad EDA industry support. Compared with the rapid growth of user member companies, the number of EDA membership kept still.

**Golden Gate Working Group**

As the Synopsys merged with Avanti in June 6, 2002 and settled the lengthy litigation Avant trade secret theft case with Cadence, the Golden Gate working group was initially formed in December 2002 by representatives from the Si2 user companies including HP, IBM, Intel, LSI Logic, Motorola, Philips, and STMicroelectronics, and EDA companies including Cadence and Synopsys. In February 2003, The OAC announced that the Golden Gate Working Group was working to build a bridge between the OA and the open Synopsys-Milkyway environments, for the shared benefit of the entire industry. The OAC’s and Synopsys’ joint goals were to define and create the bridging technology to provide interoperable flows between OpenAccess and Milkyway-based tools as well as create a smooth data transfer between IC design tools.
Because of the patent infringement suit against Cadence, another important EDA player, Mentor, had quitted the OAC 5 months earlier (July 31, 2002). The cooperation with Synopsys this time allowed the major EDA solutions on the market to interoperate, which was mandatory for building a full nanometer-level design flow, and work towards the vision of a high-performance, interoperable design environment for all EDA customers.

The release of OA 2.1

OpenAccess version 2.1 (OA 2.1) was released on May 12, 2003 to OAC. After the successful completion of the beta test of the new enhancements on the beta sites including HP, Tektronix and LSI Logic, the production release of OA 2.1 including both API specification and a high-performance RDBI was available to the public in July. A significant feature of the OA 2.1 version was the Embedded Module Hierarchy (EMH) and multi-threading support. The EMH allowed the database to maintain on demand the full relationships among the hierarchical representations of the logical, physical and occurrence views of design data throughout the flow. This enabled consistent propagation of constraints throughout the logical and physical implementation phases, provided a more efficient engineering change order (ECO) flow, and realized persistent annotations of timing, parasitic and other occurrence specific data in the design. A multi-threading support helps a CPU-intensive application to gain performance by using multiple processors and shorten the time it takes to complete the task.

With the realization of the production release, the OA 2.1 standard was rapidly becoming the API of choice for many end-user companies, EDA vendors and universities. For example, the LSI Logic’s RapidChip Platform ASIC Solution published this November (2003) was the first 130 nm design tape-out using an in-house OA based tool kit.
The appearance of OA tools

In September 2003, Cadence introduced its new Virtuoso custom design platform (version 5.033) and was made available on both OpenAccess and Cadence’s CDBA reference databases. The Virtuoso full-chip integration took advantages of the OpenAccess database to eliminate the massive file transfers otherwise needed for system-on-a-chip finishing and engineering change order flows. Cadence intended to use OpenAccess to enable seamless, bidirectional data exchange between the Cadence Virtuoso and Cadence Encounter platforms for improved productivity. But the responses from market immediately promoted the interoperability to a new level. Three days later, OAC member Sagantec, the leader in process migration and custom layout acceleration, released the Sagantec Tool Set version 7 (STS7) which included all updated version of its custom IC design products: SiClone™, SiFix™, Companion™ and Anaconda™ to ensure industry-wide open interoperability of the STS7 products with the new Virtuoso platform running on the OpenAccess database. Other members such as Mentor, PDF solution and Synchronicity also began to re-architect their tool sets to support the OA. The performance improvements are dramatic. Until November 2003, with OA support, Synchronicity Developer Suite 4.0 is proven up to 40 times faster on large, real-world design projects at 4 leading semiconductor companies.

As part of the settlement of lawsuit with Cadence the third biggest EDA company Mentor Graphics joined back OAC and announced that the Mentor Graphics’s Calibre design-to-silicon platform was integrated with the OpenAccess database in May, 2004. This integration marked a key milestone in the ongoing joint engineering development between Mentor and Cadence. The OpenAccess database became an important part of achieving full

---

7 Source: Appendix B “TOP 5 EDA COMPANIES BY MARKET VALUE 2005".
interoperability between design tools from multiple sources, allowing integrated circuit (IC) design teams to construct flows that best suit their needs.

As OpenAccess moved beyond its initial development phase and into the adoption and deployment phase, it became more important to get more companies of all sizes and interests involved in the OAC. In order to do this, in February, 2004, OAC implemented 3-Tier OAC membership structure\(^8\) to address different motive focus of members with varying degrees of responsibility and privilege, along with a new membership model. Taking advantage of this new model recently are ClioSoft, IC Manage, and Verisity.

Not only limited by the adoption, EDA startup Silicon Navigator Corp.(SiNavigator) began to create "interoperable software component" business model based on the OpenAccess. SiNavigator planed to build a platform around the OpenAccess database, turning it into a "framework" that allows startups, research groups and universities to simply write engines that plug into a database and surrounding infrastructure, as opposed to developing complete applications from scratch.

**Veto power**

On July 1 2004, Cadence handed over the veto power to the OAC change team according to its community-source agreement (June, 2001) with the Si2. The agreement stipulates that proposed changes to Genesis will be reviewed and approved by a 12-member "change team," including seven user companies and five EDA vendors. Two architects, including one from Cadence and one from a user company, may make final decisions if so delegated by the change team. From June 2001 to the end of 2002, Cadence was solely responsible for all

\(^8\) Appendix B: 3-Tier OpenAccess Coalition Membership Model.
development decisions. After that and until July 2004, the change team could not add anything new unless Cadence agreed - but since July 2004, conversely, Cadence could not add anything new unless the change team agreed.

The release of OA 2.2

In November 2004, Si2 announced the availability of the version 2.2 release of the OpenAccess (OA 2.2) specification and RDBI source code to Si2 member companies. New database support for wafer data also provides a step towards establishing strong links between design and manufacturing by eliminating the need of using the GDSII stream file format for the creation of wafer views. The new improvements extended the OpenAccess data model support further into digital design as well as manufacturing, providing significant levels of maturity and completeness. The new release included significant technology improvements defined by the OAC change team, enabling further adoption by both in-house and commercial tool providers. The 2.2 release of the OpenAccess specification and RDBI source code were open to the public in November 2005.

Semiconductor manufacturers were the first beneficiaries of the new release. OA 2.2 brought them efficient design flows that deliver accurate results in less time for larger designs. For example, by deploying OpenAccess in its mixed-signal production flow, Infineon Technologies achieved a performance gain greater than 2-fold. OpenAccess-enabled flow reduced the number of error-prone steps inherent in traditional flows. OpenAccess's ability to considerably improve tool runtime performance and cycle times along with interoperability advantage allowed Infineon's engineers to maximize design analysis rather than face ongoing data-conversion delays.
Synopsys joining the OAC

In June 2005, Synopsys Inc. dropped its resistance to join the Si2 and the OAC in response to the positive changes observed. Synopsys announced that they joined at the “director” level, meaning that the company wanted voting rights on all matters as well as participation in the OpenAccess Change Team - the collection of 12 elected companies that controlled the OpenAccess roadmap. Synopsys was not elected as a member of the Change Team, however, the company was made eligible to do so.

Cadence introduces the first OA (only) design tool

September 2006, Cadence introduced a new version of its Virtuoso custom design platform which had been around as the core technology since Cadence was founded in 1987. The biggest change with this release was that the Cadence CDBA database went away and Virtuoso became a “native” OpenAccess application. According to Cadence, from this new version to all future releases, Virtuoso will be exclusively based on the OA standard.
Appendix B: Top five EDA companies by market value 2005 (US$)

<table>
<thead>
<tr>
<th>2005 rank</th>
<th>Company</th>
<th>Market Value (12/2005)</th>
<th>Si2 membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cadence Design Systems</td>
<td>5.17B</td>
<td>OAC</td>
</tr>
<tr>
<td>2</td>
<td>Synopsys</td>
<td>2.95B</td>
<td>OAC</td>
</tr>
<tr>
<td>3</td>
<td>Mentor Graphics</td>
<td>729M</td>
<td>OAC</td>
</tr>
<tr>
<td>4</td>
<td>Magma</td>
<td>302M</td>
<td>OAC</td>
</tr>
<tr>
<td>5</td>
<td>Zuken Inc.</td>
<td>300M</td>
<td>OAC</td>
</tr>
</tbody>
</table>