After 25 years of development, the Internet has suddenly exploded and has generated great excitement. The popularity of the Internet is largely due to the influence of the WWW (World-Wide Web)\(^1\) proposed in 1989, which has made the Internet accessible to a large population\(^2\). As the spread of the Internet and the globalization of economics, the paradigm of design activity is drastically changing. Particularly, there is an increasing need for the continuous collaboration in geographically distributed design teams\(^3\).

There are some pioneering projects on the large-scale collaborative design over the Internet such as PACT\(^4\) and MADEFAST\(^5\). Design education using the Internet\(^6\) is also practiced and analyzed\(^7\). There are, however, not many observations of long-term, need-driven, real world design cases. More studies on practical designs over the Internet driven by real demands are needed.
Innovations are essential sources of the global competitive power of the industry. Many people have pointed out that it is important to increase the number of iterations of prototyping before the designers settle on a final version. The more prototypes and prototyping cycles per unit time, the more polished the final product\textsuperscript{8,9}. Hence, it is important to understand the process of distributed prototyping, with sharing design knowledge over the Internet. In this study, we will tackle the above issue by analyzing an actual design process over the Internet.

This study can be compared with protocol analysis. Protocol analysis has been used in recent years for the analysis of design activities\textsuperscript{10}. In essence, it relies on the concurrent verbal accounts (i.e. thinking aloud), sketches and gestures of one or some designers recorded on videotapes. Cross et al. say that protocol analysis is somewhere in the middle ground between the hard experimental methods of the natural sciences and the weaker, purely observational methods of the social sciences\textsuperscript{11}.

Protocol analysis of design activity fits to analyzing designer’s thought steps in detail. Among other drawbacks of protocol analysis such as the difficulty in verbalizing all the designer’s thinking and influence of the experimental setup to the data, the most serious problem is that the duration of observation is limited to several hours at most. On the other hand, the participant observation method, in which a researcher joins a design team, is closer to the methods of the social sciences. It enables the observation of design activity for several years\textsuperscript{12}. However, it is disadvantageous in that detailed records of every reason for decision-making cannot be obtained.

In this study, we chose an exemplar which satisfies the following requirements:

- It is not a given problem but is a complex and realistic design task which is driven by social requirements of the times.
- It can be observed over a reasonably long period of time.
- The prototyping cycle is visible throughout the design process.
- It is a pioneering project of an asynchronous, distributed collaboration environment\textsuperscript{13} utilizing the Internet as a new information infrastructure.

Considering the above requirements, the design process of VRML 2.0 (Virtual Reality Modeling Language, Version 2.0) specification\textsuperscript{14} was chosen as the target of this study. The VRML is a language for describing interactive 3D worlds hyperlinked on the WWW (Figure 1). One of the
#VRML V2.0 utf8

PROTO ColorBox [ field SFCOLOR color 1 1 1 ]
{
  Transform {
    children [
      Shape {
        geometry Box {}
        appearance Appearance {
          material Material { diffuseColor IS color }
        }
      }
    ]
  }
}

DEF Box_1 Transform {
  center .0 .0 .0
  children [
    ColorBox {
      color 1 0 0
    }
  ]
}

Figure 1 Snapshot of the VRML 2.0 browser

Study on the collaborative design process over the internet
keys of the design of VRML 2.0 was to unite innovations and standardization, which are often contradictory to each other.

Common Lisp is an earlier example of specification design of a computer software over the Internet (formerly known as ARPANET)\textsuperscript{15}. Early in the eighties, it was discussed by exchanging about 3000 email messages over 30 months (about 3 emails per day on average). After about ten years from the design of Common Lisp, the VRML 2.0 specification was designed on a much larger scale.

\section{VRML 2.0 specification design}

\subsection{Overview}

The design of VRML started at the VRML BOF\textsuperscript{*} in the 1st WWW Conference in May 1994 originated by the Mark Pesce’s proposals of visions and goals\textsuperscript{16,17}. The VRML Specification design was promoted and reviewed by the VRML moderated mailing list (www-vrml@wired.com)\textsuperscript{18}, which was opened in June 1994. The whole archive of the mailing list is open to the public. We have participated in the mailing list and observed it since the early stages. The first release of The VRML 1.0 specification was created by SGI (Silicon Graphics, Inc.) and based on the Open Inventor file format\textsuperscript{19}. It was finally fixed on January 29th 1996\textsuperscript{20}.

The second release of VRML added significantly new interactive capabilities. At SIGGRAPH96 (August 4, 1996) the final VRML 2.0 specification was released\textsuperscript{14}. The discussion on the mailing list is still being continued towards VRML 2.* (including a binary file format, external interfaces, internal scripting languages, etc.) and VRML 3.0 which can support multi-user scenes.

The VAG (VRML Architecture Group), consisting of eight technical experts, was founded on the 21st of August 1995, for leading and organizing the design process of VRML 2.0\textsuperscript{†}. The archive of mailing list and related materials on the VRML 2.0 design process is accessible at the Web site of VAG (Figure 2)\textsuperscript{21}. The main resource of this study is the archive of the mailing list. We can also find a complete Hypermail archive of the list at http://www.eit.com/goodies/lists/.

\subsection{Procedure}

The design of the VRML 2.0 took the following procedure\textsuperscript{23}:

- requirements,
- request for proposals,
I.3 Timeline

The earlier history and schedule of the VRML 2.0 design was as follows:

- 1994
  Nov 2, 94 VRML 1.0 First Draft

- 1995
  May 8, 95 VRML 1.0 Second Draft
  May 26, 95 VRML 1.0 First Specification

- an open debate on these proposals,
- generation of consensus around the proposal, and
- modification of the proposal.
Aug 21–23, 95 1st VAG (face-to-face) Meeting
Oct 23–24, 95 2nd VAG Meeting
Oct 24, 95 VRML 1.1 First Draft
Dec 12, 95 VRML 1.1 Second Draft

- 1996
  Jan 4, 96 VRML 2.0 Request for Proposals
  Jan 29, 96 VRML 1.0 Clarifications
  Feb 2, 96 Deadline of Request for Proposals
  Feb 5–6, 95 3rd VAG Meeting
  Feb 23, 96 Open RFP (Request for Proposals) Polling
  Mar 18, 96 Close RFP Polling (Final Polling Results)
  Mar 25, 95 4th VAG Meeting
  Mar 27, 95 VAG set up Schedule for VRML 2.0
  Apr 17, 96 VRML 2.0 Specification Draft #1
  May 30, 96 VRML 2.0 Specification Draft #2
  Jun 4, 96 VRML 2.0 Specification Draft #2b
  Jul 15, 96 VRML 2.0 Specification Draft #3
  Aug 4, 95 4th VAG Meeting (Final VRML 2.0 Specification)
  Aug 15, 96 Final VRML 2.0 Specification is available both on VAG and SGI.

In this study, we analyzed the mailing list archive over 489 days, from the release of VRML 1.0 first specification (May 1995) to the finalization of VRML 2.0 specification (August 1996).

### 1.4 Requirements
The reconfirmed requirements at the January 4, 1996 RFP are as follows:

- Performance,
- Scalability,
- Composability,
- Authoring,
- Power,
- Interoperability,
- Multi-user potential,
- Open Standard,
- Backward Compatibility, and
- Completeness.

These were the initial and rough requirements in the VRML 2.0 design process. The VRML 2.0 design was a process of refinement from the above rough set to a concrete and practical set of final specifications.
1.5 Characteristics

The characteristics of the VRML 2.0 design process can be summarized as follows:

**Open to the public**: It is open to the public on the Internet, including RFP and polling. Several face-to-face VAG meetings also played a substantial role in the process. However, the minutes of VAG meetings are also made open to the public.

**Geographically distributed**: The mailing list members are distributed widely over many parts of the world.

**Long-term**: It started in 1994 and is still continuing.

**Large-scale**: The number of the mailing list members is several thousands. 1697 people posted emails during the period of analysis. The total number of the exchanged emails were 15 051. 30.8 emails per day were posted on average (about ten times those of the case of Common Lisp).

**Diverse**: Each member has his/her own interests and concrete goals such as education, business, research, entertainment, etc. Moreover, it interacted with other design processes of the Web technologies such as Java programming language and HTML3.**

2. Analysis method

In this study, we set the smallest unit of analysis to a single email. Figure 3 shows the overall diagram of the analysis tool that we constructed for this study. As shown in this figure, we developed software tools called WWWMail, interface.cgi and Analysis. They perform text-search and index-search in an email archive from WWW browser. Figure 4 is a snapshot of the analysis tool. The procedure of analysis was as follows:

1. **WWWMail** converts the email archive in mbox format to the HTML email archive in HTML format. From an email’s header, several types of data such as date, sender’s name, email address, subject, reply information and mailer software name are extracted. Then these data are compiled to an index table. The HTML archive format of WWWMail follows the style of the HypeyMail.

2. A keyword database is created using SWISH from the HTML email archive. This database is also stored for text search.

3. Using a WWW browser, the user can search email texts by keywords through WWWAIS. WWWAIS is an interface between the search tool, SWISH, and form-capable WWW browser.
(4) A search keyword in a regular expression is sent from the WWW browser to Analysis through interface.cgi. It returns search results to the WWW browser. Analysis is a set of CGI (Common Gateway Interface) programs, written in Perl and UNIX commands such as egrep.

(5) The information filtered by the index table by time, name, email address and/or subject is created by Analysis. It is analyzed and displayed using a spreadsheet such as Microsoft EXCEL.

3 Results and discussion

3.1 Global analysis

3.1.1 Distribution

Figure 5 shows the domain distribution of all the emails posted during the period of analysis. 51.8% of the emails were posted from com domain. It shows that more than half of the total emails were posted by members of enterprises. On the other hand, the ratio of emails from edu domain was 10.4%. There were also many members from universities, research institutes and others.
Study on the collaborative design process over the internet

Figure 4 Snapshot of the mailing-list analysis tool

Figure 5 Domain distribution of emails

* .ca 4.4%, .uk 4.2%, .se 2.5%, .au 1.8%, .gov 1.6%, .jp 1.4%, .de 1.1%, .org 1.1%

Study on the collaborative design process over the internet
Detailed top 10 ranking of the domain is shown in Figure 6. Since the first prototype of VRML 1.0 specification was created by SGI based on the Open Inventor file format, the design process of the VRML 2.0 could have been continued to be led by SGI people. However, the number ratio of emails from SGI was only about 9%.

About 5% (807) of the emails came from VAG members (Figure 7). Some people pointed out that the lack of formal top-down management by a central authority would cause organizational problems. The balance between top-down management of VAG and bottom-up innovations held the key to the solution of the problems in the VRML 2.0 design process.

3.1.2 Histogram

Figure 8 shows a histogram of the number of persons (email addresses) sorted by the number of submitted emails. About half of the members posted only one mail. However, 317 members (18.7%) posted more than 20 mails. This fact also proves that the design of VRML 2.0 was fairly large-scale.

Figure 6 Top 10 ranking of domains of emails

Figure 7 Ratio of emails from VAG members

from VAG members (5%)
3.1.3 Embodiment

The monthly variation of the number of emails is shown in Figure 9. As mentioned in \textsuperscript{23}, the design process of VRML 2.0 can be divided into five phases:

- Problem structuring—a requirements phase,
- Preliminary design—a request for proposals,
- Refinement design—an open debate on these proposals,
At the highest level of abstraction, VRML is simply a file format for describing objects. Theoretically, the objects can contain anything—3D geometry, MIDI data, JPEG images, and so on. VRML defines a set of objects useful for doing 3D graphics, multi-media, and interactive object/world building. These objects are called nodes, and contain elemental data which is stored in fields and events.

†The six proposals were as follows:

- Active VRML—Microsoft
- Dynamic Worlds—GMD and others
- HoloWeb—Sun
- Out of this World—Apple
- Reactive Virtual Environment—IBM Japan

The above proposals are reviewed briefly in.


Open polling—generation of consensus around a proposal, and
Detail design—changes to that proposal to make it perfectly acceptable.

The design phase and the open polling phase overlapped each other.

Problem structuring [around Oct. 95]: The discussion on the VRML 2.0 already started before finishing the design of VRML 1.0. At the 1st VAG meeting (August, 1995), ambiguous wishes and demands were structured. Then, the basic vision of VRML 2.0 was shaped. Active and divergent arguments were developed until the announcement of RFP interacting with the design of VRML 1.1. Many innovative ideas such as PROTO node were proposed during this phase. We discuss PROTO node later.

Preliminary design [around Jan. 96]: The VRML 2.0 request for proposals was released from VAG on January 6, 1995. Six proposals were received by the due date of February 2, 1996. Exchanged emails were not many in number during this phase.

Open polling and refinement design [around Mar. 96]: After in-depth discussions, the open polling on the six candidates was conducted from February 23 to March 18, 1996. The Moving Worlds proposal developed a strong consensus. As a result, it was selected by the VRML community as the definite prototype of VRML 2.0. Discussions on the Moving Worlds proposal also started during the polling. The VRML community continued to refine the specification while examining various trade-off problems. The specifications of the nodes were refined most animatedly during this period until the publication of Draft #1 (April, 1996).

Detail design [around Jun. 96]: VAG set up and announced the schedule for finalizing VRML 2.0 design on March 27. Details of the VRML 2.0 specification were modified step by step towards the finalization on August 4, 1996. The traffic of emails was the heaviest around the publication of Drafts #2 and #3 (June, 1996).

The change log of the Moving Worlds proposal is also open to the public. 47 specifications were revised from the first proposal (January 16, 1996) to the final specification (August 4, 1996). Such prototyping cycles and incremental refinements completed the design of VRML 2.0.

3.1.4 Subject thread

The number of replies to a mail indicates a tendency of the arguments in a mailing list community. Figure 10 shows the monthly variation of the average number of emails per one subject thread. The average number of
emails in one thread was 3.33 (the total number of the threads is 4503) and didn’t vary widely.

The histogram of emails per one subject thread in June, 1996 is shown in Figure 11. Nearly half of the number of new subjects, which is 11% of the entire emails, had no reply. To the contrary, there were some threads which were followed by more than 40 emails. The duration and the depth of the discussions were diverse.

Figure 12 depicts the monthly variation of the number of subjects that had more than 20 emails. It indicates that arguments concentrated on the refinement design and the detailed design phases.
The subject threads in June, 1996 are illustrated in Figure 13. Average number of emails per one subject was 4.15, and average duration of one subject was 2.67 days. However, there were some threads whose duration was more than 2 weeks. As shown in this figure, several threads were developing concurrently in the mailing list community. Since the email is a mixture of formal and informal information, the recurrence of discussion or the mixing of unrelated topics often occurred*. Such diversity also effectively vitalizes the discussion and helps emergence of innovation.

*To avoid such problems, the following message is sent automatically at the subscription of the www-vrml mailing list:

In order to help people scan and sort the list traffic, it is recommended that all subject fields are prepended with a message type. Here is a list (by no means exhaustive) of types we’d like to use:

- **ADMIN**—List administration message,
- **ANNOUNCE**—General announcements for the VRML list,
- **APPS**—Applications for VRML,
- **EDIT**—Discussion of VRML editing tools and technologies,
- **HTML**—Issues specifically concerning VRML and HTML,
- **LANG**—Technical discussions of VRML as a language,
- **MISC**—Things that don’t fit anywhere else,
- **PHIL**—Philosophical discussion of goals and visions for VRML,
- **TOOL**—Tools and technologies for VRML authoring/editing/testing,
- **WEB**—Issues specifically relating to VRML and World-WideWeb.

Figure 12 Monthly variation of number of subjects (more than 20 mails)

Figure 13 Thread chart of subjects (June, 1996)
3.2 Detailed analysis

3.2.1 Prototyping vocabulary

As mentioned earlier, to increase the iteration in a prototyping cycle is an effective way to innovative design. However, a polysemous natural language often causes misunderstanding in communication. Hence, it is very important to share a common design vocabulary among design teams, in particular, to prototype collaboratively over the Internet. In this study, we tried to extract a prototyping vocabulary used in the VRML 2.0 specification design on the basis of the document changelog\(^7\). The prototyping vocabulary covers patterns of the operation to prototypes.

The document change log, which contains about 200 changes, for example, is written in a counter-chronological order as follows:

- **7/27/96 (re*)**:  
  - Updated Java Reference from Sony.
  - Changed Fog’s default visibilityRange from 1000 to 0 (off by default!).
  - Clarified Extrusion node—either 1 or N scale and orientation values are required. If 1, then apply to all spine pts—if N then one for each spine pt. This changed the scale defaults BACK to the original 1 1 (rather than the [1 1, 1 1] change made after Draft 3).
  - Added PLAIN style to FontStyle (was “ ”).
  - Many, many small clarifications.

The change log contains about 20 ambiguous descriptions such as:

- Many minor fixes (spelling, links, grammar, etc.), and
- Several important time-based behavior issues were clarified.

It also contains about 10 announcements such as:

- Released unofficial draft of Draft #3, and
- Offer HTML, compressed HTML and compressed Postscript versions of spec.

About 70% of the rest of such items is concerned about operation to the specification. In each piece of change, the kind of operation and the target is described with additional information.

- **[7/29/96 (re)]** Changed CylinderSensor’s offset type from SFRotation-SFFloat.
- **[7/29/96 (re)]** Moved ‘Concepts—Grouping and Children Nodes’ from 4.9.3 to 4.3.1.

*Rikk Carey, one of the VAG members.

About 12% of the emails were categorized according to the above recommendation. [ANNOUNCE], [LANG] and [PHIL] were the most frequently-used.
The prototyping feature added to VRML 2.0 enables us to create complex objects that can be reused by changing certain characteristics of the object when desired. These descriptions can be uniformly rewritten in the format of:

\[(\text{operation})\ \{\ (\text{object}),... \}\].

Then the above examples are rewritten as follows:

- (change) \{(CylinderSensor's offset type:SFRotation), (SFFloat)\}
- (move) \{('Concepts—Grouping and Children Nodes' from 4.9.3), (4.3.1.)\}
- (combine) \{(all examples), (the spec/part1/examples.html section)\}
- (restore) \{(Viewpoint description field), (non-exposed)\}
- (add) \{('Script' class), ('shutdown()')\}
- (clarify) \{(Binding rules for Background, NavigationInfo, and Viewpoint (see concepts)))\}
- (remove) \{(PointSet), (numPoints)\}
- (create) \{(MovingWorlds)\}

We have organized the operational prototyping vocabulary found in the VRML 2.0 design as summarized in Figure 14. They can be used as a starting point to construct a more general prototyping vocabulary for the distributed design collaboration.

3.2.2 Innovative process

Next, we analyzed one of the innovation processes of VRML 2.0 design in detail. VRML 2.0 features can be grouped into four main areas:

- Enhanced static worlds,
- Interaction,
- Animation and behavior scripting, and
- Prototyping new VRML object.

The first three features, which were already discussed during the VRML 1.0 design, were expected to be implemented from an early stage. On the other hand, the last feature was proposed in October, 1995 for the first time. The object prototyping feature, which makes VRML 2.0 quite charming, is

*The prototyping feature added to VRML 2.0 enables us to create complex objects that can be reused by changing certain characteristics of the object when desired.

one of the typical examples of bottom-up innovations in VRML 2.0 design. We looked at the initial design process of PROTO/USE node as an exemplar of the bottom-up innovations.

The thread that discussed PROTO/USE was ‘Why does DEF do instancing?’, including 39 emails from 22 members over 10 days (Friday, October 13, 1995–Sunday, October 22, 1995). In VRML 1.0, the DEF node is used such as: DEF s Sphere(...), in which a sphere named s is
created and added to the scene. In this thread, it was discussed whether DEF should instantiate an object and give a name to it, or should it just define a named prototype that is later instantiated when actually used.

The discussion was triggered by a message (#1* in Figure 15) questioning about the reason for instantiating rather than just defining a prototype. This first message referred to the `typedef in C++ that did not instantiate an object. Though workaround methods by displaying the first instance after translation to the desired location (#2) or by hiding the first instance from the scene (#8) were discussed, many people supported the first ques-
tion (#3, 4, 5, 7, 9, 11, 13, and 18). Among them, message #5 pointed out a situation in which a more complex object like a video stream would need versions instanciated from an abstract prototype object. Later, message #18 reinforced the advantage of prototyping by discussing a situation where the newly added behavior feature of VRML 2.0 was able to be used to create a new instance when necessary. This message proposed \texttt{PROTO} and \texttt{USE} syntax. These supporting messages were basically based on the idea of type definition and instanciation in programming languages.

A notable counter argument was made in a series of messages by a VAG member. The first message (#6) explained that a defined but unused object could not be inserted in the parsed tree of a scene graph. The second and third messages (#12 and #16) more clearly stated that VRML was not a programming language but a file format and, therefore, a VRML file must describe only a tree of instances. Message (#10) by another person also stated that to improve read/writability of VRML one should rather develop an interpreter.

At the middle point of the thread, it became clear that the point was whether VRML 2.0, whose earlier version VRML 1.0 was a rather straightforward data format, should stay on the same concept or incorporate the characteristic of programming languages. Technically, \texttt{PROTO/USE} enables the behavior node to create instances from a prototype library on the fly. Technical disadvantages of changing \texttt{DEF} to the \texttt{PROTO} semantics were that such a new feature would increase difficulty of maintaining completeness of browsers and that if \texttt{DEF} was retracted from VRML 2.0 it became incompatible with existing VRML 1.0. It was a decision process whether to put weight on the behavior node and promote the new VRML toward behavior programming. Further discussion continued between the two standpoints, until when a VAG member expressed support for the idea of \texttt{PROTO} in message #37. It led to a suggestion in message #39 to maintain \texttt{DEF} the same as before, to add \texttt{PROTO} and \texttt{USE}, and to make it capable of instanciating an object defined by \texttt{DEF} or \texttt{PROTO}.

An interesting point of this process was that the scope VRML 2.0 was supposed to cover was changed along with the evolution of the prototype. At the same time, an implicit concept behind the earlier prototype was explicit during discussion.

As summarized above, we could observe the typical innovation process through refinements of concepts and prototyping cycles in the case of VRML 2.0 design. The fact also proves that the Internet-based asynchronous, distributed, collaborative innovation is possible in practice.
4 Conclusion

This paper presented an analysis of the design process of VRML 2.0, as an exemplar of collaborative design process over the Internet. In contrast to the making process of Java programming language by a small closed expert team\textsuperscript{40}, VRML 2.0 was designed in an open process. Despite the risk of being spoiled by too many cooks, largely owing to the excellent navigation by VAG, standardization and innovative extension has successfully been materialized.

We began by analyzing the global trend of the entire process. Then we looked closely at the document change log to extract patterns of operation of the design object. We also observed the discussion process of introducing the PROTO node to the specification to examine a process of embodying an innovative idea that opened up a new dimension to VRML 2.0. We believe these analyses will serve as a reference in planning an innovative design over the Internet. The prototyping vocabulary extracted from the process can be combined with agent communication languages such as KQML (Knowledge Query and Manipulation Language)\textsuperscript{41} and studies on structuring dialogs\textsuperscript{42,*}. 

\textsuperscript{*}NCSA's HyperNews by default offers message categorization into none, question, note, mail, warning, feedback, idea, more, news, ok, sad, angry, agree, disagree\textsuperscript{43}. This set of categories, however, is supposed to be different from that suitable for organizing messages in a design process.

42 Structured Dialogs, http://union.ncsa.uiuc.edu/HyperNews/get/hypernews/structure.html

To find a generally applicable prototyping language, we need to look at processes of building a physical prototype. In such cases, issues about topology, geometrical configuration, evaluation by simulation, and manufacturing are supposed to matter\textsuperscript{7}. RACE (Research into Artifacts, Center for Engineering) has been working on the RACE (RACE Asynchronous Collaboration Environment) Project since 1995\textsuperscript{13}. In this project we are studying the process of physical prototyping and sharing knowledge by working on designs such as furniture and a micromechanism. Our experience in the RACE Project shows that knowledge sharing is the key for doing collaborative design. The hypertext structure of the Web provides us with a useful foundation for incrementally structuring knowledge created in the course of design. Since design is a process of constructing a knowledge structure out of an ill-structured problem, knowledge representation and retrieval tools on the Web including the one we built in this study are expected to serve in supporting collaborative design.

Acknowledgments

The authors are grateful to the members of RACE who contributed to this study. This study has been supported by a grant-in-aid for scientific research from the Ministry of Education of the Japanese Government.