

A Message-Based Annotation System for Collaborative Design Review

M. Witzel, G. Conti and R. De Amicis

Graphitech, Salita dei Molini, 2, 38050 Villazzano (TN), Italy

Abstract

The paper illustrates a shared annotation mechanism developed within a collaborative virtual environment for design review. This represents one of the first prototypes being developed within the European Project IMPROVE and it has been designed to be used within two distinct industrial scenarios: automotive and architectural design. The key feature of the system is that its functionalities (modelling, rendering, tracking) are distributed across a network of computers performing specific tasks and that the communication among these takes place through a standardised data message exchange. This paper shows the effectiveness of this approach by focusing on the implementation of the distributed annotation feature.

Categories and Subject Descriptors (according to ACM CCS): I.3.2 [Computer Graphics]: Distributed/network graphics

1 Introduction.

The work presented in this paper describes one of the first features implemented by the IMPROVE prototype [Imp05]. The project IMPROVE handles collaborative scenarios where several users are able to interact and conduct design review using a variety of devices such as a Head Mounted Display (HMD)/ Tiled Display System (TDS), sharing modelling and object manipulation abilities. The key feature of the IMPROVE system is the adoption of dedicated distributed components which communicate through a message passing middleware (MDM) layer.

This paper illustrates the example of a distributed annotation feature built on top of the existing infrastructure. As underlined by Hong et al. [HCC05] annotations can be considered a by-product of the user's thought and the importance of sharing annotations between members of design teams has been highlighted by a number of research [JGY02]. In our system the message distributing infrastructure is used to generate notes during the design review by the members of the panel. The review takes place in front of a high-resolution Tiled Display System (TDS) running IMPROVE which used to share and comment the design product. Each reviewer has a TabletPC running an instance of the IMPROVE system. The view of each TabletPC is synchronised with the TDS. At any time a reviewer can tap on his/her screen and move the point of view to reach a certain position. Then the user can start annotating the geometries by writing comments and sketches. This information is then distributed by the system and becomes accessible to all the other users.

The paper illustrates the implementation details and shows the strength and scalability of such approach.

2 Related Works

The way notes are traditionally taken has been studied by Melenhorst [Mel05] who compares how notes are taken using the two metaphors. The use of the sticky notes was

analyzed with respect to their function in the process. According to the author the most important function of sticky notes was to support paraphrasing information read by the user. According to the author this helped the readers to re-process the information. Always according to the author, the second function of the sticky notes was to record preliminary ideas.

As far as annotation of 3D environment is concerned, the seminal work of Loughlin et al. [LH94] introduced the post-it note concept to annotate 3D models. Following a similar idea, the authors in [JGY02] present Space Pen, a web annotation tool based on such a metaphor where the user can draw notes in the space. The note is then stored and replaced by an icon which is placed into the virtual world. The work is in turn the evolution of the authors' early work called Redliner [HW97] and [HWRB96]. In these system, the web page contains a VRML environment which can be annotated by leaving spheres attached to the geometries. These represent the notes and their content is displayed on a different frame next to the browser. Following a similar approach Tsang et al. [TFK*02] propose a novel interaction technique based on the device Boom Chameleon where the user can annotate object in the scene by drawing directly onto the 3D object. Annotation can be done by a virtual pen, or in flashlight mode by bright up parts of the model to point out specific spots. Annotations can be also stored into snapshots of the 3D environment. The possibility of annotating directly onto the 3D model has been proposed by several other authors such as [BCD01] and [JGY02].

Sonnet et al. [SCS04] propose an innovative approach to visualise annotations within interactive explosion diagrams. The approach proposed moves automatically the notes attached to the 3D objects and they are always maintained visible despite the modification applied by the user.

The authors in [TKY05] present a novel approach for overlaying annotations over the real world with respect to their current position and orientation. The annotation uses the overlay to superimpose the predefined annotations to

the AR-scene. The annotated real objects are then highlighted and if occluded, the virtual representation of that part is shown. There is no way for the user interaction with the system which becomes a plain 'presentation' tool for annotations. Similarly in [KKS*00] the authors describe how annotations are displayed using an image overlay at various points of the environment on wearable computers. Annotations are placed in the appropriate positions on panoramic images in augmented reality.

Hong et al. [HCC05] show the use of annotations for 3D electronic books. The annotation, which is collaboratively shared with other users, is made through use of multi-resolution textures blended into the scene.

Ramachandran et al. [RK03] propose an annotation system for web pages which uses a standardized XML based data format. In the current prototype implementation, the inking coordinates and all the attributes and properties needed to store ink annotations are stored in the local client machine as a separate annotation layer. Whenever the browser loads a URL the layer is dynamically overlaid on the rendered document. The inked points, text ranges, relative reference positions and other annotation attributes, such as window sizes and time stamps, are stored along with the URL of the annotated page in an annotation XML schema.

Finally some authors [KMM*02] adopt annotations as a way to impart an aesthetic touch to rendered 3D models. The approach proposed by the authors is especially designed for artists and to let them annotate a 3D model as they would do in reality by painting over the object's surface.

3 The System Architecture.

The IMPROVE system consists of five distributed, autonomous subsystems: Communication Backbone, Interaction, Tracking and Rendering Component and a central repository. Each subsystem, which can be physically located on a separate machine, communicates with the other modules through a high-level message exchange.

Each component is deployed entirely autonomously of the other system components. This gives the system a high degree of flexibility with respect to the physical distribution of the devices as well as robustness. By specifying consistent well-defined protocols, the components are made independent and replaceable. The number of IMPROVE clients is in fact only restricted by the channel capacity of the communication backbone server.

The *Communication Backbone* connects all components via network and uses XML messages to provide the data exchange which can be categorized into modelling, annotation, configuration, calibration and shape data. Each software component declares which data they are interested in by subscribing to various channel topics each defining a specific data set. In the annotation case the communication backbone creates two channels the 'client' and 'server' message which specify the name of the client generating the annotation and the server broadcasting the data. The details of this mechanism are described in later sections of the paper.

The responsibility of the *interaction component* is generally related to model/scene/session management,

shape management, view management, annotations, design modifications, light creation and control and it follows a publisher/subscriber approach. Interactions can be distinguished into two different kinds: the first is applied locally; the second is additionally transmitted via the communication backbone to the other components. Annotations can be captured in the form of text, voice, or drawing small sketches in vicinity or on the surface of the model. They define various status fields for unsolved, priority, type, etc..

The *tracking component* sends the information captured by the different devices supported and it wraps them into normalized XML messages called "IMPROVE Entities" and it sends this data via the client channel through the communication backbone.

The *rendering component* visualizes the virtual models and annotations using a variety of display devices.

Finally the *repository* implements the functionality to store and retrieve invoked commands and it stores configuration data like calibration and setup data. This allows for maintaining a history of actions and a versioning of modifications applied to the model. Each component implements its own interfacing client to interact accordingly with the networked components. Objects associated to bodies during a session are equally stored in the meta-information structure of a certain model such as annotations. Repository entities like models and annotations can be modified or updated by a certain user, temporarily blocking access for other session participants. By using a subscriber/publisher approach, the run-time session status (e.g. locking status) can be easily retrieved through real-time processing of the review process.

4. Collaborative Navigation and Annotation of Entities

Traditional design review sessions are undertaken by having a peer controlling the information exchange. Usually there is little or no possibility for the participants to interrupt the process and to pay attention to a particular issue and then to rejoin the reviewing group at a later point in time.

The scenario envisaged by the project is illustrated in Fig. 1. As shown on the right side of the picture a panel of reviewer is standing in front of a large scale Tiled Display System (TDS). The navigation is performed through a SpaceMouse shared by the peer reviewer. Each reviewer can interact with a local instance of the system through a TabletPC.

The solution developed by the authors allows each participant to independently navigate and annotate the objects in the scene at any time. The annotation is done by scribbling over the TabletPC's screen. This allows for a greater freedom during the design review, and it lets users annotate information in large numbers which in turn stimulates them to communicate ideas and comments while discussing the overall project.

An additional option to this scenario is illustrated on the left side of the Fig. 1 where an additional user is running a further instance of the system within an Augmented Reality setup.

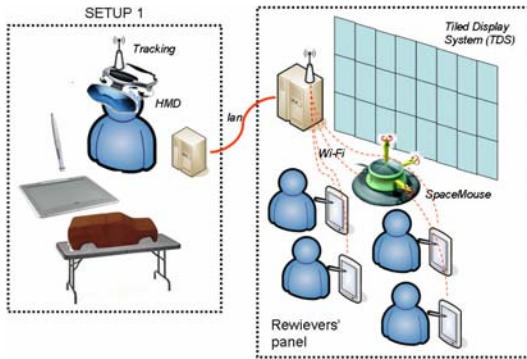


Fig. 1: the scenario where the annotation system is used

As soon as the user creates a new annotation, by tapping and holding the pen over an object, a translucent whiteboard is displayed on top of the scene. This is implemented through a transparent bitmap with an alpha channel that covers the whole model on the screen. The user is then able to sketch over this panel.

Since each participant to the session uses a TabletPC (TPC) whose position is not tracked, it is of utmost importance to be able to provide a natural form of interaction necessary to navigate within the scene with a simple pen. This is necessary once the user's view is decoupled from the position shown on the TDS and he/she starts to adjust the current view to reach a more advantageous viewpoint.

The 2D-to-3D interface is specifically designed for use with TPC (see Fig. 2) i.e. based on one-button actions without the need for further buttons to be used. In fact, assuming the use of a simplest TPC pen without any button within its bezel, the only action available to the user is the pen touching the screen.



Fig. 2: a view of the 2D-to-3D interface specifically developed for use with the TabletPC

As soon as the synchronization is interrupted the system running on the TPC enters the annotation mode where the user can then select each geometry and make annotations over it. During this phase there is obviously no tracking system therefore the only pointing device is the pen used by the TPC. Since the only information available is its position over the screen there is no reason to show a 3D view of the pen. As soon as this mode is activated the user can start annotating the scene. As far as the annotation

process is concerned two distinct operational modes have been identified: a synchronous and an asynchronous review.

4.1 Synchronous Review.

The IMPROVE system offers this mode for a conventional approach to design review: one peer reviewer broadcasts viewpoint and interaction events to the passive participants of the distributed review session. Thus, the viewpoint is shared on all display devices (clients) and this allows for a collaborative discussion by the participants.

4.2 Asynchronous Review.

At certain points during the review, a design review participant wants to scrutinize a certain model entity in-depth. Going out-of-sync from the main review process, the participant is enabled to temporarily leave the group by changing his/her view to take annotations over the model of interest. While the group continues the normal review, the single user keeps on receiving all interaction messages and tracking data like so that the view transformations can be synchronized back at any time.

During this phase since the input devices are purely 2D-based (pen, mouse), a picking filter needs to decide what action will be invoked.

- a) Clicking in void space results in a movement of the transformation control
- b) Clicking on a base marker opens an annotation on the overlay for modification
- c) Clicking on an entity will create a new annotation. It worth noting that this way a 1:n relation between entity and annotation is realized. In other words the note is uniquely attached to the specific geometry together with the information on the current point of view. In fact, starting from the assumption that while creating an annotation the entity of interest is in clear view of the participants, this information can be used later to replace an optimal point of view which allows a comfortable reading of the note. This is done at later stage using linear spherical interpolation to changing smoothly the current view transformation's values to those stored along with the annotations' data.

As mentioned annotations are taken following a post-it metaphor i.e. the note is represented through a panel, attached to the geometry, facing the user. For the geometry to be visible the note is placed next to the current geometry and rendered in overlay to the scene's image. As soon as it is created, the system automatically includes the user name, which is extracted from the system's login information, and a timestamp. On a dedicated area, the user is able to stroke 2D sketches with the pen (see Fig. 3). Next to the note there are buttons for accepting and rejecting the changes as well as to minimize, or else to close, the annotation.

When done, the user goes back to synchronous mode by using the unlock button at the bottom of the display and he/she ideally rejoins the group by getting in sync with the TDS' point of view. The newly created annotation is stored in the repository and it is broadcasted to all connected clients within the distributed review.

4.3 Data Flow XML Message Passing.

As illustrated the various subsystems are connected by the communication backbone. This is implemented using a message-oriented middleware. Thus, it has been necessary to agree upon a specification of the message exchange. Within the IMPROVE system all data to be exchanged are wrapped by well-defined messages called “IMPROVE Entities” (IE). The data to be exchanged are organized into various topics, i.e. annotation, shapes, configuration data, which enables the clients to filter at the earliest time possible which messages are of interest. The messages are sent through two channels, a client channel transmits messages dedicated to the repository. Similarly a server channel broadcasts messages to all clients who interpret the received message. They then take the appropriate actions. For instance, although the exchanged messages have a fixed structure, an annotation is rendered differently on the TDS (3D) and on a TPC (2D). In the case of annotations, a mechanism needed to be implemented which prevented the concurrent editing of the same annotation by multiple users.

To prevent the creator from receiving the very same annotation, the IMPROVE entity has also a local ID tag associated with it.



Fig. 3: (top) an example of annotation. (Bottom) three minimized notes with different status.

4.4 Annotation Creation.

The message exchange protocol for the creation of new annotation is straightforward: first, an update message carrying the specification of the new annotation, marked with the id=-1 (i.e. new) is sent via the client topic channel to the repository (see Fig. 4). There, the IMPROVE Entity is processed and stored in the repository. Then, the repository assigns a new unique ID to the annotation and it sends it to the clients which subscribed to the server channel topic.

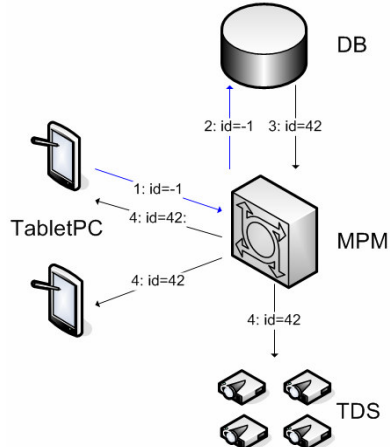


Fig. 4: message flow during the creation process

4.5 Entity Locking.

A locking mechanism for entities needed to be implemented to prevent the simultaneous editing of annotations. Each client keeps track of which annotations are currently locked. By knowing if an annotation is not locked, a lock request is sent out to the repository via the client topic channel. The lock is in turn accepted by the repository component by sending out an acknowledgement via the server channel topic to inform each client. Each subscribe of the server topic channel keeps track of which annotations are currently marked for modification (i.e. they are in the “locked” status). The “un-locking” is handled analogously and for the sake of brevity it is omitted.

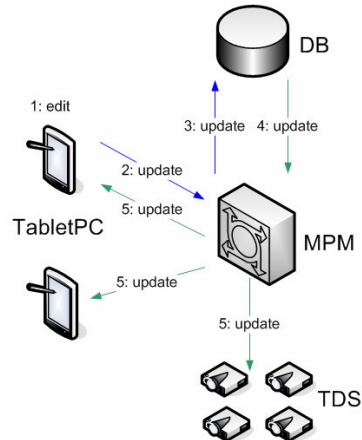


Fig. 5: message flow during the creation process

4.6 Annotation Update.

When an annotation has been edited on one IMPROVE client and previously locked on the repository, the new annotation is sent out to the previous protocol and the

updated annotation is broadcasted by the *repository* to all subscribed components (see Fig. 5).

5. Discussion and Future Work.

The result of the first prototype have been shown in Fig. 2 and these prove the potential of such distributed architecture. In fact the annotation prototype, which requires only simple data transmission has already proved to be a valid support for a number of data such as:

- scribbling data,
- user information,
- view transformation,
- locking mechanism,
- storing/reading to/from the *repository*

This represents only the first step of the ongoing development which aims at creating a flexible framework whose data structure is capable of adapting to the requirements of 3D data exchange by supporting complete transmission of geometrical information through the broadcasting of commands to be interpreted by the underlying geometric modeler.

Further extensions to this very system are being developed for supporting voice annotations as well as for attaching documents. Finally current annotations do not yet support the annotation of model entities with geometric sketches directly in the virtual space to capture design modification intentions.

6. Acknowledgements

The work presented in this paper is part of the IMPROVE EU-Project.

References

- [BCD01] Bourguignon, D., Cani, M-P. and Drettakis, G: Drawing for Illustration and Annotation in 3D, In *Proceedings of EUROGRAPHICS'01* (2001), 114-122.
- [HW97] Hanson A. J., Wernert E. A.: Constrained 3D Navigation with 2D Controllers. In *Proceedings of the Conference on Visualisation '97*, (1997), 175-183.
- [HWRB96] Harmon R., Patterson W., Ribarsky W., Bolter J.: The Virtual Annotation System. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, (1996), 239-245.
- [HCC05] Hong L., Chi E. H., Card S. K.: Annotating 3D electronic books. In *Proceedings of CHI - Conference on Human Factors in Computing Systems* Portland, OR, USA (2005), ACM Press, 1463 – 1466.
- [Imp05] The Improve Project. Available at <http://www.improve-eu.info/>. Last accessed 22 Dec. 2005.
- [JGY02] Jung T., Gross M. D., Yi-Luen Do E.: Sketching annotations in a 3D web environment. In *CHI '02 extended abstracts on Human factors in computing systems Minneapolis, Minnesota, USA* (2002), ACM Press, 618 – 619.
- [KMM*02] Kalnins R. D., Markosian L., Meier B. J., Kowalski M. A., Lee J. C., Davidson P. L., Webb M., Hughes J. F., Finkelstein A.: WYSIWYG NPR: drawing strokes directly on 3D models. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques, San Antonio, Texas*, (2002), ACM Press, 755 – 762.
- [KKS*00] Kourogi, M., Kurata, T., Sakaue, K., Muraoka, Y.: Improvement of panorama-based annotation overlay using omnidirectional vision and inertial sensors. In *Proceedings of The Fourth International Symposium on Wearable Computers* (2000), ACM Press, 183 – 184.
- [LH94] Loughlin M. M., Hughes J. F. An Annotation System for 3D Fluid Flow Visualisation. In *Proceedings of the Conference on Visualisation '94, IEEE*, (1994), 273-279.
- [Mel05] Melenhorst, M.: Observing professionals taking notes on screen. In *Proceedings of International Professional Communication Conference*, (2005), IEEE Computer Society, 540 – 545.
- [RK03] Ramachandran, S., Kashi, R. : An Architecture for Ink Annotations on Web Documents. In *Proceedings of Seventh International Conference on Document Analysis and Recognition* (2003) 256-260.
- [SCS04] Sonnet H., Carpendale S., Strothotte T.: Integrating expanding annotations with a 3D explosion probe. In *Proceedings of the working conference on Advanced visual interfaces, Gallipoli, Italy*, (2004), ACM Press, 63 – 70.
- [TKY05] Tenmoku, R.; Kanbara, M.; Yokoya, N.: Intuitive Annotation of User-Viewed Objects for Wearable AR Systems. In *Proceedings of Ninth IEEE International Symposium on Wearable Computers* (2005), ACM Press, 200 – 201.
- [TFK*02] Tsang M., Fitzmaurice G. W., Kurtenbach G., Khan A., Buxton B.: Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. In *Proceedings of Symposium on User Interface Software and Technology, Paris, France* (2002), ACM Press, 111 – 120.