



Workshops der
Wissenschaftlichen Konferenz
Kommunikation in Verteilten Systemen 2009
(WowKiVS 2009)

Towards an Interest Management Scheme for Peer-based Virtual
Environments

Florian Heger, Gregor Schiele, Richard Süselbeck, Christian Becker

10 pages

Towards an Interest Management Scheme for Peer-based Virtual Environments

Florian Heger¹, Gregor Schiele², Richard Süselbeck³, Christian Becker⁴

¹ florian.heger@uni-mannheim.de

² gregor.schiele@uni-mannheim.de

³ richard.sueselbeck@uni-mannheim.de

⁴ christian.becker@uni-mannheim.de

Chair of Information Systems II
University of Mannheim, Germany

Abstract: A fundamental task in peer-to-peer-based Massively Multiuser Virtual Environments is providing all peers with a consistent view of the environment. To do so, state changes must be propagated to peers. To limit the resulting network traffic, existing approaches often restrict the distribution of a state change to peers for which it is relevant. The identification of such peers is known as *Interest Management*. Typically Interest Management is done based on so-called *Areas of Interest*. An Area of Interest is a spatial area around a user's avatar. If a change occurs inside this area, it is relevant and must be reported to the corresponding peer. We propose to extend this approach with so-called *Areas of Effect*. An Area of Effect specifies the spatial area of the virtual environment that is directly influenced by a given state change. A state change is propagated to all peers whose Areas of Interest intersect with the change's Area of Effect. This allows us to model complex state changes with arbitrary and possibly dynamic influence areas. They may even affect multiple areas at once. In this paper we describe our approach and give an overview on the current state of its implementation.

Keywords: MMVE, Peer-to-Peer, Interest Management, Area of Interest

1 Introduction

Massively Multiuser Virtual Environments (MMVEs) allow the interaction between thousands of users in a shared virtual environment. Users are represented in the MMVE by their avatars, virtual entities that are controlled by the users and act on their behalf. In order to maintain the illusion of a credible virtual environment, the users constantly have to be provided with updates about state changes. A lot of information has to be propagated, which results in a high volume of network traffic and induces the need for sophisticated filtering schemes. These schemes determine the information which is relevant for every user. Network traffic is optimized by only propagating relevant information. The design and the implementation of such schemes are generally known as *Interest Management* [Mor96]. Because scalability of an MMVE very much depends on optimization of network traffic, the implementation of an effective Interest Management scheme is a key factor for creating scalable MMVEs with millions of concurrent users.



This is even more challenging for peer-to-peer (P2P) based MMVEs, like [HL04] [KLXH04] [CYB⁺07] [YV05], in which the management of the virtual world is distributed over all participating devices. Such MMVEs have become increasingly popular in recent years, due to their large potential with respect to better scalability and lower operating costs.

Our goal is to provide an Interest Management scheme for P2P-based MMVEs that enables highly scalable and dynamic systems. We base our approach on the well-known concept of *Areas of Interest* (AoI) and extend it with a novel concept which we call *Areas of Effect* (AoE). An AoI specifies a spatial area around a user's avatar in the virtual environment. This area describes the user's interest in his environment. The relevancy of information is determined based on this area: A peer only receives updates about state changes inside the AoI. In addition to this, an AoE specifies the spatial area in the virtual environment that is affected by a given state change. Thus, a change can affect only a specific point, e.g. in case of a slow movement of a virtual object, or a larger area in the MMVE, e.g. a line in case of a very fast movement. This allows us to model complex state changes with a single update instead of a series of updates, reducing the amount of updates that must be distributed. In this paper we propose our initial approach for scalable Interest Management in P2P-based MMVEs based on AoIs and AoEs. We present AoEs in more detail, show their potential and possible shortcomings and discuss how this concept can be used in an Interest Management scheme. This work is part of the Peers@Play project [www].

The rest of this paper is organized as follows: In [Section 2](#) we describe our system model and discuss requirements for an Interest Management scheme. After that we discuss related work in [Section 3](#) before we present our approach in [Section 4](#). Finally, we give an overview on the current implementation state of our scheme, draw conclusions and give a preview on our future work in [Section 5](#).

2 System Model and Requirements

Before discussing existing approaches for Interest Management in P2P-based MMVEs, we first describe our target system environment in more detail. After that we present a list of requirements which a suitable scheme must fulfill.

Our system model consists of a number of users that want to use an MMVE, their devices, a communication network and the MMVE software. Users may be located at any place in the world. The number of users is a priori undetermined and can change dynamically. A device executes the MMVE software and is connected to a common communication network, e.g., the Internet. We call such a device a *peer*. A peer can join and leave the system at any time. Each user is represented in the MMVE by a special character, called *avatar*. To interact with the MMVE, the user controls his avatar to perform different *activities*. Activities can also be initiated by so-called *non-player characters*. If an activity is executed, the system creates *events* describing the resulting state changes and propagates them to other peers.

To be usable in such a system, an Interest Management scheme must fulfill the following requirements. We adopt these requirements from [Mor96] and adapted them to P2P-based systems.

1. *Restriction to Relevance*: An Interest Management scheme has to determine only the relevant information for each peer. Relevant means all information which is needed to guarantee a correct game-play and to maintain the authenticity of the virtual world in the users'

perception.

2. *Consistency*: An Interest Management scheme further has to provide all users with enough information to guarantee a consistent world state on each peer. Because consistency very much depends on users' perception, there is information which is more and information which is less important to guarantee consistency.
3. *Distribution*: In P2P-based MMVEs there is no central entity. The filtering of information has to be done in a decentralized manner. Hybrid forms, e.g. MMVEs using zones with coordinators, are possible.
4. *Density Awareness*: In MMVEs users tend to cluster at certain locations, e.g. inside cities. An Interest Management scheme has to take that into account and has to adapt to these clusters in order to propagate updates efficiently.
5. *Interest Scope*: The interests of users are not necessarily symmetric. For example one user may see another user but not vice-versa, because one user is hidden behind an obstacle and watches the other user. Some capabilities are also directional, e.g. the range of view. Two users are able to see each other except one is behind the other. An Interest Management scheme has to take the asymmetry as well as the direction of interest into account.

3 Related Work

Various proposals have been made on Interest Management in MMVEs. Most proposals either use visibility or spatiality as a filter to determine the relevant updates. An overview can be found in [BKV06].

Steed et al. [SA05] use visibility as a filter. They propose a network partitioning scheme called *Frontier Sets*. The environment is sub-divided into cells. Frontier Sets are computed and used to determine if users are visible to each other. Updates are only propagated to the visible users. Frontier Sets have an advantage inside rooms and buildings. Outside the scheme shows weaknesses because of the resource-intensive calculation of Frontier Sets.

Proposals using spatiality as a filter often include the concept of *Areas of Interest* (AoI), e.g. [KLXH04] [HCC06] [BHS⁺08] [YV05]. The user is represented in the virtual environment by an avatar. The AoI is a space around the avatar describing the limitations of its sensing capabilities. Updates are filtered based on the AoI. The dominant spatial structure of the AoI is circular, e.g. [HCC06] [BHS⁺08]. Hu et al. [HCC06] propose AoIs which change size dynamically depending on the number of peers inside the AoI.

The concept of AoI goes back to the proposal of a spatial model of group interaction in virtual environments by Benford et al. [BF93]. In this model every object in a virtual environment is surrounded by an *aura*. Objects carry their aura with them when they move through the environment. The environment monitors for aura collisions. When auras collide, interaction is possible. Benford et al. also mention that an object typically has different auras (e.g. differentiated by size and shape) for different media. It may be possible that in a large space without obstacles one user can see another user but can not hear him, because the visual aura is larger than the audio aura.

The spatial model includes the concept of *focus* and *nimbus*, which is further described in [GB95] and [SKH02]. Auras are divided into focus and nimbus. Focus represents a user's perception, nimbus represents an object's perceptivity. A user is aware of an object if its focus intersects with the object's nimbus.

Filtering using spatiality is further done by dividing the environment into *regions*, e.g. [KLXH04] [HCC06] [FRP⁺08] [CYB⁺07] [YV05]. Each region is managed by a central entity, e.g. by a server or a coordinator node. Updates are propagated by the central entity. A user's interest into a region is determined depending on the location of the user (e.g. [KLXH04]) or the intersection between the user's AoI and the region (e.g. [HCC06]). In these approaches, scalability very much depends on the right partition and the right choice for coordinator peers.

An often used model for update propagation is the *Publish-Subscribe Model*, e.g. [KLXH04] [HCC06] [CYB⁺07] [YV05]. When a user is interested in a region, he subscribes to the central entity and from that point on receives updates about state changes inside that region. When the user loses his interest, he unsubscribes from the region. The Publish-Subscribe Model has the disadvantage of creating additional network traffic for the handling of subscriptions and unsubscriptions when the users move.

Frey et al. [FRP⁺08] break with the Publish-Subscribe Model: Bounding boxes around the avatars are computed and information is exchanged directly between avatars that may potentially collide with each other based on these boxes. They do not mention a distinct entity for state changes. The propagation of updates depends on the avatars.

In [GLZ05] GauthierDickey et al. describe an algorithm for event ordering in peer-to-peer games. The peers are organized in an n-tree. They propose to model state changes explicitly as events. Events are described as a tuple consisting of an action, the action's location and a function representing the action's scope of impact. The scope of impact is used to determine which event has to be propagated to which peer. Their algorithm allows events to be propagated quickly between peers which are located near to each other in the virtual environment. Even they model state changes as events, they do not represent events as first level entities inside the MMVE.

We think that proposals using the concept of AoI can be enhanced concerning the AoI's spatiality and the representation of state changes. The sensing capabilities of users are influenced by user and environmental properties, e.g. there are users which have a larger range of view and users which are hidden behind obstacles. Therefore, the AoI's spatial structure and size should be changed dynamically. In contrast to [HCC06], we think that the change should depend on user and environmental properties, not the number of surrounding users.

In MMVEs there are state changes which have properties, e.g. state changes which influence a certain area or remain for a period of time inside the MMVE. By representing state changes as entities in the Interest Management scheme, the changes' properties can be taken into account for update propagation.

According to [GB95] and [SKH02], our approach includes areas complementing the AoI and Interest Management is done depending on the intersections. In contrast, the areas of our approach represent new first level entities describing the state changes instead of being tied to existing entities like users or objects. These first level entities are not only characterized by spatiality, they are also characterized by further properties like time and priority. In the following, we describe our approach.

4 An Interest Management Scheme for Peer-to-Peer-based MMVEs

Our Interest Management scheme is based on the concept of AoI management. We modify the concept and extend it with the concept of so-called *Areas of Effect* (AoE). In our scheme, state changes are modelled as first level entities, so-called *events*. In case an activity is executed that induces a change in the virtual environment, update events have to be issued. Events allow us to model even complex state changes which affect certain areas or last for a certain period of time by controlling their distribution. Events are represented by AoEs. An AoE is the event's spatial extension in the virtual environment. It's properties resemble the event's characteristics.

By representing events by AoEs, we intend to optimize the amount of updates that has to be sent over the network. Using explicit entities for complex state changes allows us to propagate information about the event in a single update and compute possible changes in the following without a need for further updates. For example the information about the emergence of rain clouds is sent only once in a single update to all affected peers. In the following, changes are computed locally on every peer according to environmental circumstances, e.g. the wind.

In the following, we give details about AoI and AoE. Then we describe how relevant updates are determined. Finally we describe our system architecture and give an overview on the current implementation state.

4.1 The Area of Interest

We mentioned before that the AoI's spatial structure depends on the sensing capabilities of the users. These capabilities are influenced by user and environmental properties. These properties can change, e.g. the environment that surrounds a user changes when the user moves. Therefore, we do not define a distinct spatial structure in our Interest Management scheme. We also include the possibility to change the structure and it's size dynamically according to changes of these properties.

Our approach is based on the assumption that an appropriate spatial structure for the AoI is equivalent to finding a trade-off between complexity of the structure's computation and accuracy of update propagation. Possible structures range from a rectangular structure to an arbitrary formed geometric structure. While a rectangular structure means easy computation at the expense of an unaccurate propagation, a free formed structure means complex computation at the benefit of an accurate propagation.

Benford et al. [BF93] discussed that there should be more than one AoI regarding the different kinds of media, e.g. there should be different AoIs for viewing and hearing. Following this, we do not limit our scheme to a single AoI per user.

Instead, for every user there is an indefinite number of AoIs. All AoIs surround the user. They are represented by indefinite spatial structures. If the user moves on, the AoIs also change location according to the user's movement.

4.2 The Area of Effect

The AoE is the representation of an event inside the MMVE. It has a spatial structure, a lifetime and a priority. The spatial structure represents the event's impact on the environment. Because

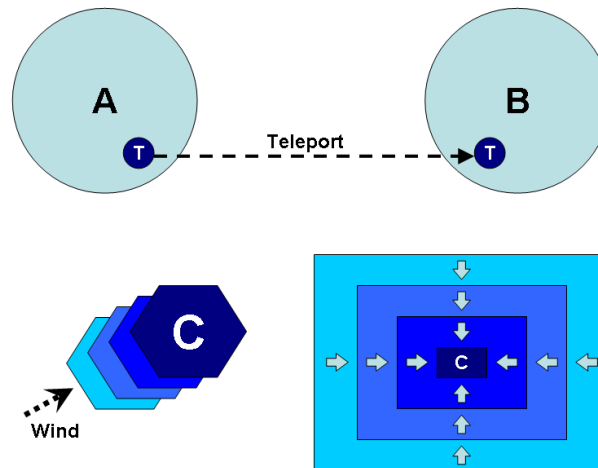


Figure 1: AoEs of a teleport (top), an accumulation of clouds represented by a hexagon (bottom left) and a rectangle (bottom right).

structure and size depend on the event's characteristics as well as environmental properties and both factors may vary from event to event, the AoE has an indefinite spatial structure. The AoE also may exist for a certain lifetime and its structure may change during lifetime according to environmental properties. The information about priority is needed to determine the importance of events in case of overload, in order to detect the first events to drop.

In the current version of our scheme we concentrate on simple geometrical forms, e.g. circles, rectangles and points. Based on our requirements analysis we find that geometrical forms are a fit for most events in an MMVE. They can be computed in an efficient way and, therefore, are a desirable solution. An example for an AoE given as a single point is a movement event. A line could be used for a very fast movement. An example for an event that affects a wide area in the environment is an accumulation of clouds which causes rain in a certain area. In our scheme the corresponding event may be represented by a circular, elliptic, rectangular or hexagonal structure. Even though we concentrate on geometrical forms right now, we also think about using free formed structures in the future. Free formed structures offer the possibility of a very accurate representation of an event. The main challenge concerning these structures will be to find an efficient way of computing them. In addition to this, the spatial structure of AoEs can also be split into multiple parts. MMVEs often include events that affect multiple non-adjacent locations in the virtual environment, e.g. if a user is teleporting from one place to another (see [Figure 1](#) top). This event demands the propagation of updates to users at both locations. The AoE representing the teleport consists of two parts, both of them have the spatial structure of a point.

Because AoE's have a lifetime, our scheme is able to represent events that affect the environment only for a single moment as well as over a period of time. The spatial effect of events

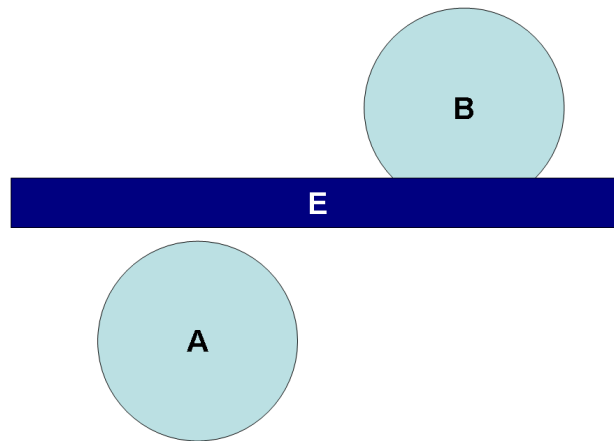


Figure 2: The determination of relevant updates based on AoEs.

can change during lifetime. For example the accumulation of clouds mentioned before may change its location according to the wind (see [Figure 1](#) bottom left). Also the size of the accumulation may change according to changes in weather conditions (see [Figure 1](#) bottom right). The accumulation of clouds represented by the hexagon changes location according to the wind. Another accumulation represented by a rectangle changes size according to changes in weather conditions.

The priority of an AoE is used to decide whether the propagation of updates can be passed on in case of a high user density (Requirement 4). For example, the rain caused by the accumulation of clouds mentioned before may not be propagated to users which are crowded together at a place. The rain is used to create a better atmosphere and is not an integral part of game-play and therefore has a low priority. At the moment, this is still work in progress and not included in our prototype, yet. We expect this concept to be a main topic of our future research.

4.3 Determination of Relevant Updates

To determine whether a given event is relevant for a certain user, we compute the intersection of the event's AoE and the user's AoI. A user is provided with updates about an event, if its AoI intersects with the AoE of the event. If there is an AoI of user A and an AoE of event E, then the information about E is propagated to A if A's AoI intersects with E's AoE. [Figure 2](#) shows an example. The AoI of user B intersects the AoE of event E. Therefore, B has to be provided with updates about E. The AoI of user A does not intersect E's AoE and no updates are propagated.

If an AoE consists of multiple parts, updates are propagated to all users whose AoI intersects with at least one part of the AoE. If the AoI of a user intersects with multiple parts, the update is propagated only once. Assuming that the AoI approximates the users' interest in updates in an appropriate way, the determination of relevant updates in our scheme fulfills Requirement 1 (*Restriction to Relevance*). Requirement 5 (*Interest Scope*) is not fulfilled by our scheme at the



moment and will be part of our future work.

4.4 System Architecture

The main architectural question when realizing this approach is where the computation of intersections between AoIs and AoEs is done. In a P2P system there are two main possibilities: Either there is one central instance for Interest Management, i.e. a single peer, or Interest Management is done in a decentralized manner, e.g. determination of relevant updates is done on each user's computer. Using one central instance has the advantage of a good consistency control (Requirement 2). There is also only one instance that has to be provided with updates about events. The main disadvantage of that architecture is the possibility of overloading the central instance. In that case, the central instance acts like a bottleneck for the whole architecture. Decentralized Interest Management offers the advantage of avoiding the possible bottleneck, but makes it difficult to maintain consistency. Also each user's computer has to be provided with information about the positions of all other peers' AoI, in order to compute the intersections of the areas in a decentralized manner.

Because both architectures offer advantages as well as disadvantages, we propose to use a hybrid architecture including aspects of both. We propose to tessellate the environment into zones. Each zone is administrated by a coordinator peer. The coordinator peers are responsible for Interest Management. This architecture avoids the bottleneck and still offers the possibility to control consistency in a partly centralized manner. It also fulfills Requirement 3 (*Distribution*). The development of our hybrid architecture is still in progress and a closer description will be part of our future work. Nevertheless, we give a short description of the propagation algorithm in the following.

Our algorithm is based on two assumptions: (1) For every zone there is a coordinator peer responsible for Interest Management. This peer is dynamically chosen from all peers that participate in the MMVE. (2) Every peer is able to determine the coordinator peers and their corresponding zones. In the peers@play project we use a Distributed Hash Table (DHT) for this lookup. Updates about an event are propagated as follows:

1. In case an event is triggered on a peer, the information about the corresponding AoE is sent to the coordinator peer of the zone in which the peer is located.
2. The coordinator peer determines which zones overlap with the AoE. This is determined based on the maximum spatial expansion of the AoE.
3. The coordinator peer sends the event and its AoE to each coordinator peer which handles an overlapping zone.
4. Each coordinator peer then computes the intersections between the AoIs of the users inside his zone and the AoE.
5. Finally, the information about the event and its AoE is propagated by the coordinator peer to the affected peers.

4.5 Implementation State

We have implemented a basic version of our Interest Management scheme and integrated it into the Peers@Play framework. The computation of intersections between AoIs and AoEs as well as the propagation of updates depending on the outcome of the intersection operations are already implemented and working stable for a single zone. The current version of the implementation includes variably sized circular AoIs and AoEs. Currently, we are working on implementing additional geometric forms and multi-part AoEs.

5 Conclusions and Future Work

In this paper we presented an early view on our proposed Interest Management scheme which adopts the widespread concept of Areas of Interest and extends it with Areas of Effect. Based on a requirements analysis we proposed to model state changes as explicit entities (so-called events). Areas of Effect are the spatial extension of events in the virtual environment. By modelling state changes as entities with distinct properties we are able to represent complex events whose properties change over time. We provided a first glimpse on our algorithm for update propagation and an overview on the current implementation state.

The implementation and evaluation of our approach is currently on the way. A main topic of our future research will be to find a solution on how to handle users that cluster at certain places (Requirement 2). Event priorities are one possible approach that we will employ in this area. We are also going to develop an event model for describing the characteristics of complex state changes in order to ease the definition of events and their corresponding Areas of Effect.

Bibliography

- [BF93] S. Benford, L. Fahlen. A Spatial Model of Interaction in Large Virtual Environments. In *Proceedings of the 3rd European Conference on Computer-Supported Cooperative Work*. 1993.
- [BHS⁺08] J. Botev, A. Höhfeld, H. Schloss, I. Scholtes, M. Esch. The HyperVerse - Concepts for a Federated and Torrent-Based "3D Web". In *Proceedings of the 1st International Workshop on Massively Multiuser Virtual Environments at the IEEE Virtual Reality*. 2008.
- [BKV06] J.-S. Boulanger, J. Kienzle, C. Verbrugge. Comparing Interest Management Algorithms for Massively Multiplayer Games. In *Proceedings of the 5th ACM SIGCOMM Workshop on Network and System Support for Games*. 2006.
- [CYB⁺07] L. Chan, J. Yong, J. Bai, B. Leong, R. Tan. Hydra: A Massively-Multiplayer Peer-to-Peer Architecture for the Game Developer. In *Proceedings of the 6th Annual Workshop on Network and Systems Support for Games (Netgames)*. 2007.
- [FRP⁺08] D. Frey, J. Royan, R. Piegay, A.-M. Kermarrec, E. Anceaume, F. L. Fessant. Solipsis: A Decentralized Architecture for Virtual Environments. In *Proceedings of the 1st*



International Workshop on Massively Multiuser Virtual Environments at the IEEE Virtual Reality. 2008.

- [GB95] C. Greenhalgh, S. Benford. MASSIVE: A Collaborative Virtual Environment for Teleconferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 2:239–261, 1995.
- [GLZ05] C. GauthierDickey, V. Lo, D. Zappala. Using N-Trees for Scalable Event Ordering in Peer-to-Peer Games. In *Proceedings of the 15th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. 2005.
- [HCC06] S.-Y. Hu, J.-F. Chen, T.-H. Chen. VON: A Scalable Peer-to-Peer Network for Virtual Environments. *IEEE Network* 20:22–31, 2006.
- [HL04] S.-Y. Hu, G.-M. Liao. Scalable Peer-to-Peer Networked Virtual Environment. In *Proceedings of the 3rd ACM SIGCOMM Workshop on Network and System Support for Games*. 2004.
- [KLXH04] B. Knutsson, H. Lu, W. Xu, B. Hopkins. Peer-to-Peer Support for Massively Multiplayer Games. In *Proceedings of the 23rd Conference of the IEEE Communications Society (INFOCOM)*. 2004.
- [Mor96] K. L. Morse. Interest Management in Large-Scale Distributed Simulations. Technical report, Department of Information & Computer Science, University of California, Irvine, 1996.
- [SA05] A. Steed, C. Angus. Supporting Scalable Peer to Peer Virtual Environments Using Frontier Sets. In *Proceedings of the IEEE Virtual Reality*. 2005.
- [SKH02] J. Smed, T. Kaukoranta, H. Hakonen. Aspects of Networking in Multiplayer Computer Games. *The Electronic Library* 20:87–97, 2002.
- [www] The Peers@Play Project: <http://pap.vs.uni-due.de>.
- [YV05] A. Yu, S. T. Vuong. MOPAR: A Mobile Peer-to-Peer Overlay Architecture for Interest Management of Massively Multiplayer Online Games. In *Proceedings of the 15th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. 2005.