

Tabletop Interface

In order to reach a co-located situation-sensitive IT-support, a tabletop interface based on multi-touch technology was designed. By using multi-touch gestures as input and rigid body dynamics as simulation-mechanism of real life physics, an easily understandable and realistic interaction is provided.



Figure 4: Multi-Touch Tabletop-Interface

Text input is realized using virtual onscreen keyboards, to ensure interoperability with other interfaces.

The tabletop is especially promising for applying creativity-techniques that afford increased communication and coordination of the participating users. This could be due to very specific problems requiring multi-disciplinary actors or due to the character of the technique itself.

Contact



Technische Universität München



Marc René Frieß

E-Mail: friess@in.tum.de



Florian Forster

E-Mail: forster@in.tum.de

Technische Universität München

Chair for Applied Informatics/Cooperative Systems
Boltzmannstr. 3
85748 Garching
Germany

Web: <http://www11.in.tum.de/forschung/projekte/creative>

IdeaStream

Collaborative Creativity

The IdeaStream application supports you and your team in finding innovative approaches to solve problems by applying computer-supported creativity techniques.

IdeaStream was initially developed by Florian Forster as *generic creativity support system* and proof-of-concept for both the software architecture and a *unified model* of the creative process.



To take into account the different situations in which creativity can happen, several user interfaces, reaching from distributed to co-located work, were added. Additionally, the system was attached to an innovation-management platform for empowering its use within companies.

Process Model

IdeaStream is built around an abstract process model. The core design concept of the model is based on the observation that a creative process is typically a sequence of *convergent* and *divergent* phases. In divergent phases, ideas can be generated and edited. In convergent phases, previously generated ideas can be evaluated.

A combination of phases into a sequence allows the system to model and instantiate several different creativity processes and creativity techniques such as:

- (Anonymous) Brainstorming
- Assumption Reversal
- Brainwriting 6-3-5
- Morphological Analysis
- Osborn checklist
- Random Stimulus Technique

Customized techniques, specifically adapted to a particular context or company, are also possible by using this model.

The system hereby acts as a facilitator, guiding the participants in the creative process as it is intended by the respective creativity techniques.

Multi-Interface Support

The software architecture of IdeaStream particularly addresses the design principle of „*exchangeability of user interfaces*“. A schematic overview of the architecture is given in figure 1.

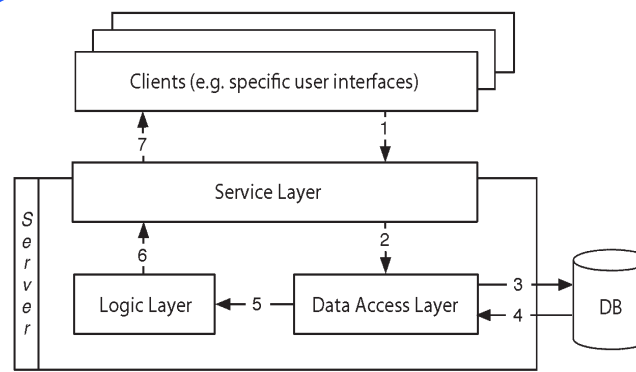


Figure 1: Architecture of IdeaStream

The service layer is accessible to arbitrary clients via Hessian webservice. The Hessian protocol is a lightweight binary web-service protocol that builds upon HTTP.

Because it makes use of its own custom binary format instead of XML for object serialization, the average message length is much shorter and objects can also be serialized/unserialized significantly faster on both the client and server system. There is also a wide range of implementations of the Hessian protocol for different programming languages such as PHP, C++, C# or Java.

The multi-interface approach also allows for a concurrent and mixed use of different interfaces.

Web/Mobile Interfaces

As IdeaStream was originally directed towards a distributed usage, the first implemented interface was especially designed for classical web-browsers. Most central element is a (shared) virtual whiteboard, allowing real-time, synchronous and parallel work via AJAX.

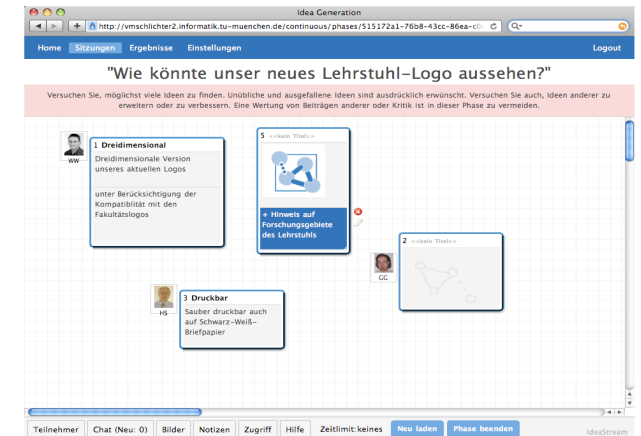


Figure 2: Web-Interface

Ideas within IdeaStream are typically represented as cards, containing a set of figurative or textual aspects. Collaboration and social awareness in a distributed setting is supported by a chat and visual feedback mechanisms. To enable a broader range of possibilities of distributed usage, another interface addresses mobile clients, especially mobile web-browsers with JavaScript support. The example shown in figure 3 is directed to the iPhone.



Figure 3: iPhone-Interface