

animation with blocks of text, ExMS exploits the same expressive mechanism as emoticons. Combining a textual message with a semi-imagery representation of a face adds new layers of meaning, e.g. irony, guides the recipients' interpretation of the message, and expresses the sender's emotional state or character. Creating animated emoticons that move and transform over the temporal flow of the message will merge modalities in similar ways to cartoon and animated film.

With this theoretical framework in mind, we identified the following design challenges: How to allow users to compose highly situated and expressive avatar-based messages in a time-efficient manner? What is the relation between gestures and movements and the surface skin/appearance of the avatar? The interaction descriptions given below have been realized in a mock-up concept demonstrator.

2. Design Choices

2.1 Composing Messages: Moods and Events

In order to support simple expressive messaging, we decided to center the composing around two axes, which we called moods and events. Moods relate to the basic emotional undertone in the message. Composition starts by selecting from a palette of moods, e.g., happy, distressed, angry and busy. The animation showed, supposedly reflects this mood in a simple cartoon-like fashion. Depending on the length of the total ExMS - decided by the sender - the animation would make continuous loops until the end. In the preview mode the user can, at any time in the composition phase, inspect the message composed so far.



In addition to mood animations, avatars should be able to perform simple events, e.g. jump, dance, smile, laugh, and weep (possibly connected to sound effects). Events are intended to mark and enforce certain text snippets, or break the basic mood undertone of the message. Events can be added at specific points in the message. This is done through previewing the message and then, at the appropriate moment, interrupting the message where the event should be inserted. From the list of events, one is chosen, which brings the user back into preview mode again, this time with the event inserted in the loop break closest to the point where the user interrupted the message.

Since all mood and event animations start and stop at a neutral position frame, continuity of movement can be ensured without involving the user.

Combining text and animations in appropriate temporal constellations will be central in conveying punch lines or other emotional content. In the mock-up, however, such support is not well articulated. It allows only simple text input, which distributes itself over the length of the message. In the final implementation this has to be further investigated.

On an algorithmic level, such temporal juxtapositions also have to be supported. To this end, we have sketched a standard for encoding expressively enhanced messages: the Mood Markup Language or MoodML (Figure Fehler! Unbekanntes Schalterargument.). MoodML is a sketch for a SGML-type markup language for elaborating a textual message with text-global mood information and specifically positioned animated events. It is designed to minimize bandwidth, in that MoodML text messages can be transmitted between sender and rendering server at little cost, and to allow for exact timing of animation events in text.

Composition is executed through loops of inserting events/texts and reviewing the message. When the user is satisfied with the work, he sends it away via his telephone book.

```
<MoodML skin="88F2A0F" id="janet"
  to="0405047123" mood="happy;8">
  I'm excited to see you!
  <event="jump">
  Be there at ten.
  <event="dance">
</MoodML>
```

2.2 Quickies and Recycled Messages

In some cases, standard scripted responses to messages can be useful. Our design gives users the possibility to save the script from both incoming and outgoing ExMSs as 'quickies'. Such quickies can be reused later, exchanging the balloon text to fit the new situation. Quickies enable users to share particularly spectacular or useful sequences of moods and events. Reusing other friends' messages as quickies is possible through the saving of the message MoodML script. When I want to reuse the quickie for my own avatar, whose skin might be very different from my friend's, the system identifies the name of the animation in the MoodML script and then re-instantiates the sequence in the skin of my avatar. This takes us to the issue of skins.

2.3 Skins – avatar appearance

Although the temporal juxtaposition of text, moods and events constitutes one channel of expressivity, naturally the animations or the surface appearance of the different animations add, transform and personalize that expressivity. The purely graphical design of the avatars we call skins. Any mood or event can thus be represented in any type of skin that reflects a personality feature, identity or trait preferred by the user. Again, skins may not reflect features that the user necessarily possesses, but rather one which supports the kind of expressivity the user favors.

Before initiating the ExMS service, users should be able to choose a prefabricated skin from a library. Each skin will represent moods and events through varying styles, in addition to attributes such as age and gender. In order to ensure personalization and individualization of the avatars, it should also be possible to complement a skin with a set of add-ons such as clothing style, colors, T-shirt texts, hats and even tools and instruments such as walkmans, sticks and roses. Most users will probably make use of standard skins included in the service package. Other skins may be produced professionally for marketing purposes as a product placement platform: users can get new skins for free, if they allow their avatar to wear a brand emblazoned T-shirt.

Some artistic users, however, will want to manufacture their own skins. To this end, open standards and APIs are absolutely essential. No single organization will have the

