# An approach to Contextual Transformation in EBSs

Suhas Chikkanna

## 1. Background

Among Distributed Systems, the Event-based Systems(EBSs) are used to disseminate event notifications on the occurrence of an event, generated by an event producer. The disseminated event notification is delivered to the event consumer, which subscribes to an event of particular interest. The event notifications are delivered to the event consumers based on a match/satisfying criteria between the subscription and event notification. The EBSs are predominantly used in the supply chain, new ticker and share market. These applications are widely distributed, with heterogeneous components making up a system-wide architecture. An event detected &/or generated by a component, is propagated through event-notification service to the event consumer. The current publish/subscribe systems utilizes a common context, to infer an event produced by an event producer and, subscription subscribed by an event consumer. The producer and consumers may often differ in their respective contexts. And the differing contexts between producers and consumers leads to a difference in the interpretation of the event. Below in the related work section, ACTrESS suggests a solution to this differing context.

## 2. Related Work

ACTrESS suggests automatic contextual transformation for differing contexts in Event-Based Software Systems at runtime, transparent to end users and also, ACTrESS overcomes major challenges related to contextual transformations. In ACTrESS, the Producer Context is transformed to Root Context (at the broker the producer/publisher is connected to) and the Root Context is transformed to Consumer Context at the fringe of the broker network(i.e., at last broker the consumer is connected to).

## 3. Problem Statement

The ACTrESS prototype implements greedy algorithm and works best in most cases. But in ACTrESS, there may result unnecessary work in spite of producer and consumer being in the same context. For instance, in a channel-based pub/sub system and Topic-based pub/sub system. And also, certain topologies such as mobile networks where the clients(i.e., producer/consumer) move from one broker to another, the same transformation (Producer Context  Root Context  Consumer Context) performed in the former broker has to be performed again on the latter broker, where the client may newly join. We try to address this with our heuristic in the below section.

## 4. Our Approach

Our approach focuses on, contextual transformation with respect to taking into account the number of transformations to be performed and the number of messages generated in the network due to the transformations. Our approach proposes to perform contextual transformation, at the first broker with a subscriber in different context. Below listed are the different cases in order to illustrate our approach.

**Listing 1.** Example usage of the listing package

```
class S {
    int f1 = 42;
    public S(int x) {
            f1 = x;
    }
}
```

### 4.1 Citations

Use citations to refer to other papers (**??**) and books (**??**).

### 4.2 Tables

Table 1 shows how a table looks like.

| English | German |
|---|---|
| cell phone | Handy |
| Diet Coke | Coca Cola light |

**Table 1.** Translations.

### 4.3 Figures

Figure 1 shows a simple figure with a single picture and Figure 2 shows a more complex figure containing subfigures.


**Figure 1.** SOLA logo.


(a) TUDaLogo (b) SOLALogo

**Figure 2.** Two pictures as part of a single figure through the magic of the subfigure package.

### 4.4 Source code

The listings package provides tools to typeset source code listings. It supports many programming languages and provides a lot of formatting options.

Listing 1 shows an example listing. Code snippets can also be inserted in normal text: `\lstinline|int f1 = 42;|` gives `int f1 = 42;`

### 4.5 Miscellany

**Capitalization.** When referring to a named table (such as in the previous section), the word *table* is capitalized. The same is true for figures, chapters and sections.

**Bibliography.** Use `bibtex` to make your life easier and to produce consistently formatted entries.

**Contractions.** Avoid contractions. For instance, use "do not" rather than "don't."

**Style guide.** A classic reference book on writing style is Strunk's *The Elements of Style* (**?**).

## 5. Another Section

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

## 6. Yet Another Section

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

## 7. Conclusion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.