The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)

# A NOVEL APPROACH TO CONTEXT PREDICTION IN UBICOMP ENVIRONMENTS


Stephan Sigg
Chair for communication Technology
University of Kassel
D-34121 Kassel, Germany
sigg@uni-kassel.de

Sandra Haseloff
Chair for communication Technology
University of Kassel
D-34121 Kassel, Germany
shaseloff@comtec.eecs.uni-kassel.de

Klaus David
Chair for communication Technology
University of Kassel
D-34121 Kassel, Germany
comtec@uni-kassel.de



## Abstract

The ability to predict future contexts significantly expands the possibilities of context-aware computing applications. However, an incorrect prediction may also mislead the application and may result in inappropriate application behaviour. We study influences on the prediction accuracy and propose a novel approach to context prediction in ubiquitous computing environments. In our paper we introduce a context time series prediction algorithm based on local alignment techniques. Our approach has the potential to improve the prediction accuracy since it explores the observed context history in more detail than current algorithms. In conclusion, we present simulation results that support our studies.


## I. Introduction

Context awareness plays an increasingly dominant role in modern software systems. It is the attempt to provide applications with relevant information on the surrounding context [1]. Several definitions of context have been given in the literature. In our work we adapt the definition given by Dey.

**Definition 1**
*Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves [1].*

The information to characterise the situation of an entity is supplied by various sensors. These sensors may provide contextual information about the place, time or the physical situation of the user. In ubiquitous environments the utilisation of third-party sensors that are provided by nearby devices or objects additionally extends the number of available sensors and hence makes context aware applications more powerful. Mobile phones nowadays incorporate a multitude of sensors. First of all the location of the user may be computed by the cell-id or by GPS information. Additional sensors such as a microphone or a camera are typically available. Some phones also possess sensors to measure light intensity. The signal strength or SIR may provide location information like indoors/outdoors. Additional nearby sensors can be accessed by Bluetooth, infra red or WLAN which possibly also provide useful details on resources within reach. Furthermore software sensors like a calendar or a clock might provide meaningful information.

A context-aware application may, for example, infer that a customer is in a business meeting and accordingly redirect incoming calls to his mailbox instead of initiating a resource devastating paging request.

One promising contemporary approach to context awareness is proactivity or prediction. Context prediction opens a wide variety of possibilities for novel applications. An application situated on a mobile device may track personal habits and predict future contexts of the user even if they are spontaneous. A network operator might, for example, improve network throughput if the number of customers in every cell is known for the near future. For active users this may be achieved by predicting the future positions of mobile customers based on their observed position, speed and direction.

But also other behaviour patterns can be explored. If it is a common habit of a user to watch the games of his favourite soccer team on his mobile device, or if he is about to start his daily video conference with his wife, the system may proactively reserve resources by shifting other calls to neighbouring cells.

Some work has been done on predicting future contexts based on the observed context history. In these approaches, the context history is analysed for typical patterns. Studies on human behaviour suggest that such patterns exist (see for instance [2, 3, 4, 5]). Based on this knowledge and on the current context time line, future contexts are forecast. The authors of [6] and [7] for instance study prediction based on GSM cell location histories. To describe the transition probabilities from one location to the other they use a Markov predictor and a weighted graph respectively. In [8] Unix-shell commands are predicted by a simple pattern matching method that forecasts the next command based on the frequency of occurrence of the last two succeeding commands. A work considering context prediction with arbitrary sensors is given in [9]. The authors present the idea of a context diary that stores observed contexts and a method to predict future contexts based on the context diary. A revealing overview on context prediction is given by Mayrhofer in [10]. Mayrhofer proposes an architecture for context prediction and indicates some benefits and challenges thereof. Mayrhofer chooses an approach called growing neural gas to predict arbitrary future contexts.

Regardless of which algorithm is utilised, the prediction accuracy is crucial to any application that employs context prediction. We will in Sect. C. study impacts on the prediction accuracy and give guidelines to aid the application designer. On the basis of our results we propose an architecture for context prediction in Sect. III. that has the potential to improve the prediction accuracy.

Contexts may be comparable by various mathematical operators. All approaches stated above only utilise few available operators (in most cases the operator '$=$'). This may be sufficient for some contexts like, for example, location but is definitely ignorant of further context information. Since we typically know few about a context it is not clear why one would not utilise additionally available mathematical operators like, for example, '$<$', '$\cdot$' or '$-$', since this may enable the use of a stronger context prediction algorithm. Our proposed architecture supports prediction methods that utilise arbitrary mathematical operators. Our approach to context prediction is not restricted to specific context types as for example location, but may process arbitrary context data.

All approaches mentioned base prediction on aggregated high-level contexts. In our opinion this is the suggesting approach to the task of context prediction but it also introduces new challenges (cf. Sect. **??**). Our proposal is to base prediction directly on the low-level context information and apply the context aggregation afterwards.

A further issue seldom mentioned in the field of context prediction is the naturally fuzzy information basis: Context series that represent human behaviour patterns. Unlike machines, humans tend to alter their habits and typically are not exact regarding the duration of a context. Typical prediction methods are easily misled by these small changes in behaviour patterns. This problem of minute variations in human behaviour is therefore typically not addressed in studies on context prediction. We present a method that is capable of abstracting from slight variations to some extent.

## II. Preparative Discussion

In this section we introduce elementary concepts that are frequently used throughout our work.

### A. *Frequent Patterns in Human Behaviour*

In the research branches context-awareness and context-prediction, researchers assume that the behaviour of a user contains distinguishable patterns by which a context or even a time series of contexts can be deduced.

This assumption has to be made with care since the sensor output is only one part of the definition of a certain context. The mood of the customer may, for example, have considerable influence on the context even though it can hardly be measured by a sensor [11]. Also, as the authors in [12] state, the sensor output leading to a specific context may change over time for one user and may even completely differ among different users. On the other hand, numerous authors indicate the existence of typical patterns in many fields of human behaviour. As [2] states, behaviour consists of patterns in time. For instance, the authors of [3] observe typical behaviours in teamsport games. It is even possible to recognise the software programmer of a piece of programming code based on his programming style [4].

For our purposes we assume that typical patterns exist in human behaviour that can also be observed in context pattern created by sensor measurements.

### B. *Context Types*

We distinguish between high-level context information, low-level context information and raw sensor data. The output of any sensor is considered as raw data since it most probably needs further interpretation. Different manufacturers produce sensors with varying output even if the sensors are of the same class. This is because of possibly different encodings of the sensed information or due to a different representation or accuracy. Two temperature sensors may for instance differ in the unit (Celsius or Fahrenheit), in the measurement accuracy or in the measurement range. A preprocessing of raw sensor data is necessary so that further operation is not influenced by special properties of the sensor values themselves.

In this context acquisition step raw sensor data is normalised to a representation utilised by all further context processing steps. We say the data has become low-level context information. The low-level context information of two arbitrary sensors of the same class measured at the same time in the same place is, apart from a possibly differing measurement accuracy, identical, provided that both sensors are in good order. The output of all temperature sensors may, for example, be normalised to degree Celsius.

In order to obtain high-level context information the low-level contexts are further aggregated with low-level contexts of other sensors. From low-level contexts describing the temperature, light intensity and the humidity it might be possible to infer the high-level context outdoors/indoors. There is no limit to the level of aggregation. Several high-level contexts may be aggregated to again receive high-level context information. A common application or service expects high-level context as input data since the application designer takes no interest in the composition of high-level contexts from low-level context information but in the high-level context the user is currently in.

### C. *Context Prediction Schemes*

In the literature context prediction is usually based on high-level context information (see for instance [6, 7, 8, 10, 13, 14]). However, the prediction based on high-level context information has vital restrictions due to a reduced knowledge about the context itself.

#### 1) *Reduction of Information*

Some of the information contained in a low-level context is lost when transformed to a high-level context since the transformation function is typically not reversible. Once we abstract from low-level context information to high-level context information we cannot unambiguously obtain the low-level contexts we abstracted from. This is true even if we know the transformation function. A trend in a low level context time series may be masked in the more general high-level context time series. Assuming, for example, that the high-level context $H_1$ (eg. outdoors/indoors) is inferred if the low-level context $L_1$ (eg. degree Celsius) falls below 20 and $L_2$ (eg. lux) is above 1000. However, given $H_1 =$ 'outdoors', we only possess the somewhat fuzzy information $L_1 < 20$ and $L_2 > 1000$.

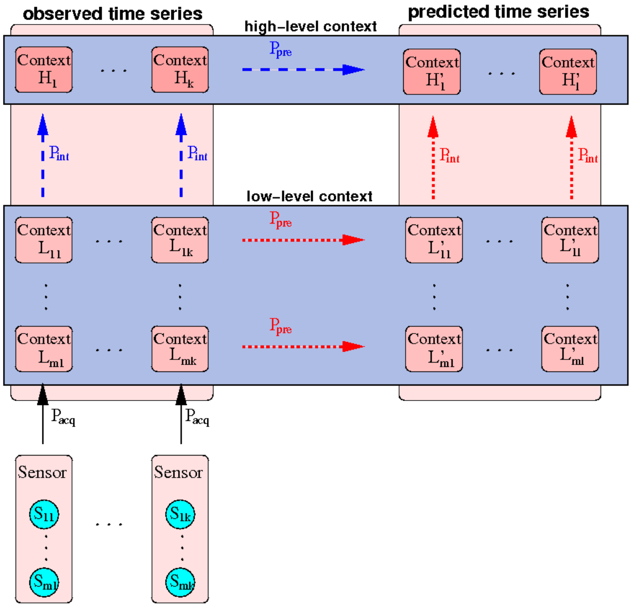


Figure 1: Dotted arrows: Prediction based on low-level contexts. Dashed arrows: Prediction based on high-level contexts.

#### 2) *Reduction of Certainty*

We assume that the prediction method bases its decision on a context history of $k$ time series elements. In case of low-level context prediction each of the time series elements is composed of $m$ low-level contexts. For high-level context prediction each time series element is represented by one high-level context. According to the order in which the context prediction and context interpretation steps are applied, the prediction accuracy may be decreased when prediction is based on high-level context information (cf. Fig. 1)). In the figure, the output values of all $m$ sensors attached to the architecture at time $t_1 \dots t_k$ are denoted by $S_{11} \dots S_{mk}$. These are then transformed to low-level context information in the context acquisition step. The resulting low-level contexts are denoted by $L_{11} \dots L_{mk}$. The dotted (dashed) arrows describe the order of actions in case of prediction based on low-level (high-level) contexts. $H_1 \dots H_k$ therefore denote the inferred high-level context time series after the context interpretation and $H'_1 \dots H'_l$ ($L'_{11} \dots L'_{ml}$) denote the high-level (low-level) context time series after the context prediction has been applied. Let $P_{int}$ and $P_{acq}$ in the figure be the probabilities that no error occurs in the context interpretation and the context acquisition step respectively. We write $P_{pre}(i)$ if we want to address the probability that the context element at time $t_i$ is predicted correctly. If the prediction is based on low-level contexts, the probability that the predicted context element at time $t_i$ is correct is

$$P_{\text{low-level}}(i) = P_{\text{acq}}^{m \cdot t} P_{\text{pre}}^{m}(i) P_{\text{int}} \quad . \tag{1}$$

If the prediction is based on high-level contexts, the probability that the predicted context element is correct is

$$P_{\text{high-level}}(i) = P_{\text{acq}}^{m \cdot t} P_{\text{int}}^{t} P_{\text{pre}}(i) \quad . \tag{2}$$

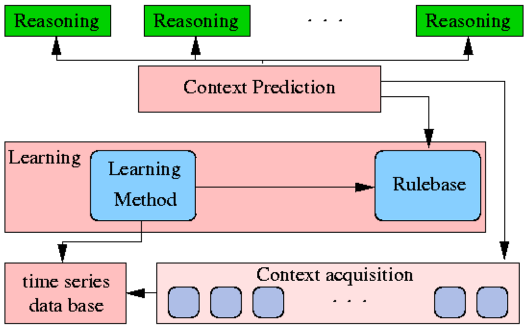


Figure 2: Our proposal of a general architecture for context prediction.

Comparing these results we obtain

$$\frac{P_{\text{low-level}}(i)}{P_{\text{high-level}}(i)} = P_{\text{pre}}^{m-1}(i) \cdot P_{\text{int}}^{1-t} \quad . \tag{3}$$

Designers of context prediction architectures therefore have to consider the ratio of prediction to interpretation accuracy, the number of low-level contexts involved and the size of the context history considered. Keep in mind that we have ignored the possibility to correct accidentally an error in one step by another error in a following step, since the accuracy gain thereof does not outweigh the decreased comprehensibility.

## III. Architecture for Low-level Context Prediction

Prediction in our proposal is based on low-level context information (cf. Fig. 2). The reasoning modules and the context acquisition are not part of the architecture.

### A. *Time Series Data Base*

We define the notion of a context time series.

**Definition 2**
*An unary context time series is a series of consecutive events in time that are associated to one specific sensor. A general context time series is a union of several unary time series whose events share the same timestamps.*

Note that we have restricted ourselves to discrete events. We approximate continuous signals by taking many consecutive samples in a short period of time.

For each sensor attached to the data base we track a unique ID and a time series of low-level contexts. We further track with every sensor ID the maximum and minimum value discovered.

### B. *Learning Module*

The learning module is composed of a learning method and a rule base. The rule base contains the rules (time series in our case) that guide the prediction module.

The learning module constantly monitors the time series stored in the time series data base and eventually uses some or all of these to construct and update rules in the rule base.

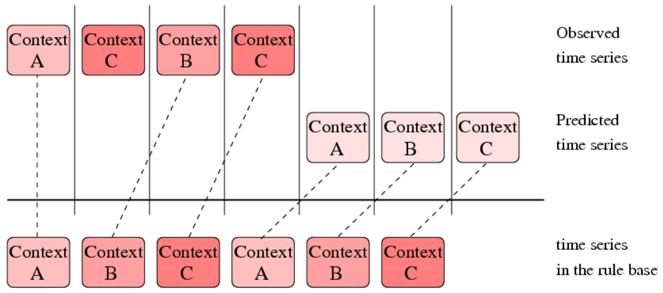


Figure 3: Context prediction by alignment methods.

Our learning method adds a new observed time series to the rule base every time this time series is not considered similar to any time series in the rule base. The similarity is expressed by an alignment rating of the optimal alignment between these two time series (see Sect. 2)) [15].

### C. Prediction Module

The actual prediction is done by the prediction module. This module accesses the rule base and the momentary sensor data provided by the context acquisition module. We propose a local alignment prediction procedure. The prediction based on local alignment techniques is schematically described in Fig. 3. In the figure we observe that the method requires at least one typical time series that is stored in the rule base as well as the sequence of recently observed contexts. The alignment method searches for the subsequence in the typical time series that is most similar to a suffix of the recently observed series of contexts. Since the time series in the rule base is considered typical we assume that the suffix of the recently observed context series is followed by the same context pattern the subsequence in the typical context series is followed by.

In order to rate the similarity of two time series we measure the similarity between these according to a distance metric. The distance metric compares single entries in the time series. The similarity of two time series is estimated as the sum of the similarities of single entries.

#### 1) Comparison of Single Time Series Entries

A single entry in a time series $S = s_1 \dots s_m$ contains sensor information from various sensors. For $i \in [1 \dots n], j \in [1 \dots m]$ we normalise every sensor value $v_i \in s_j$ to $v_i' \in [0, 1]$. An entry in a time series is then represented by a point in a $n$-dimensional coordinate system where $n$ is the number of sensors involved. Each sensor output is uniquely mapped to one axis of the coordinate system (cf. Fig. 4). Note that we require that two time series can only be compared if the sensors involved match in number and type, since we compare the corresponding points in the n-dimensional hyperspace. The correlation between any two points in this hyperspace is given by their Euclidian distance.

#### 2) Alignment of Time Series

The following definitions are taken from [15] and are adapted to our notation.

| | $v_{\max}$ | $v_{\min}$ | $v_i$ | $v_i \cdot (v_{\max} - v_{\min})^{-1}$ |
|---|---|---|---|---|
| $S_A$ | 10 | 0 | 5 | 0.5 |
| $S_B$ | 5 | $-3$ | 3 | 0.375 |
| $S_C$ | 7 | 2 | 1 | 0.2 |

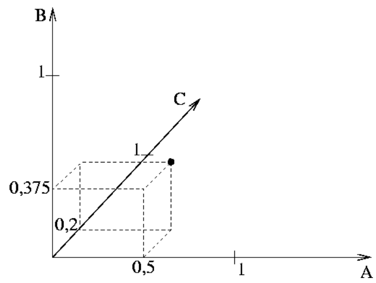


Figure 4: Top: Example input values for three sensors and the corresponding normalised values. Bottom: Coordinate system describing the configuration of the three sensors.

**Definition 3**
*An alphabet $\Sigma$ is a set of symbols $\sigma$ where each symbol uniquely matches a configuration of sensor values.*

Every time series of length $n$ can be represented by a string $\sigma_1 \dots \sigma_n$ with $\sigma_i \in \Sigma$ where every time series element is described by a symbol $\sigma_i \in \Sigma$.

**Definition 4**
*Let $s$ and $t$ be two time series over the alphabet $\Sigma$. And $\{-\} \notin \Sigma$ a gap symbol. We denote the empty time series with $\lambda$. Let $\Sigma' = \Sigma \cup \{-\}$ and $h : \Sigma'^* \rightarrow \Sigma^*$ a homomorphism with $\forall \sigma \in \Sigma : h(\sigma) = \sigma$ and $h(-) = \lambda$. An alignment of $s$ and $t$ is a pair $(s', t')$ of strings over the alphabet $\Sigma'$ so that the following conditions hold:*

1. $|s'| = |t'| \geq \max\{|s|, |t|\}$
2. $h(s') = s$
3. $h(t') = t$
4. *there is no position where both $s'$ and $t'$ have a gap.*

Obviously an alignment is not an exact matching of context patterns but a fuzzy matching of patterns that comes most closely to the exact matching. To be able to rate the quality of alignments we introduce an alignment rating.

**Definition 5**
*Let (s,t) be an alignment of two time series $s$ and $t$. An alignment rating $\delta : \Sigma^* \rightarrow \mathbb{R}$ is a metric describing the similarity between the time series $s$ and $t$.*

Since we search for subsequences in context time series that are maximally similar according to the alignment rating we further modify the description given above.

**Definition 6**
*Let $\delta$ be an alignment rating with the optimisation aim minimisation. A local alignment of two strings $s = s_1 \dots s_m$ and $t = t_1 \dots t_n$ is an alignment of substrings $s' = s_{i_1} \dots s_{i_2}$*

Table 1: High-level to low-level context prediction accuracy.

| Interpret. error | Ratio of pred. accuracies |
|---|---|
| 0.1 | 1.073 |
| 0.2 | 0.732 |
| 0.3 | 0.634 |
| 0.4 | 0.547 |
| 0.5 | 0.036 |

*and* $t' = t_{j_1} \dots t_{j_2}$ *for s and t. An alignment* $A = (s', t')$ *of the substrings* $s'$ *and* $t'$ *is an optimal local alignment of s and t if* $\delta(A) = \max\{d(s', t') | s'$ *is substring of* $s, t'$ *is substring of* $t\}$. *In this formula* $d(a, b)$ *is the distance metric of the local alignment.*

Provided we have a distance metric, we are able to compare two time series by searching for their optimal local alignment. We choose the sum of the Euclidian distance between all pairs of time series elements in the alignment as our distance metric. The alignment rating of the optimal local alignment is calculated as the sum of the distances between the subsequent pairs of time series elements in the optimal local alignment.

## IV. Simulation Scenario

We experimentally verify the results from Sect. II. with Foxtrot (Framework for cOnteXT awaRe cOmpuTing), which is currently developed by our research group [16]. We implemented the prediction algorithm described in Sect. III. for low-level and high-level prediction modules in Foxtrot . The context interpretation and acquisition modules do not actually interprete or acquire context data but apply a predefined error probability.

We utilise four synthetic sensor output patterns of 41 elements each. In the course of the experiment we repeatedly choose one of these patterns at random. To exclude side effects, the learning process of the context prediction algorithm is disabled. For the low-level and high-level prediction algorithms the input patterns are identical. We further modify the context interpretation error in several simulation runs from $0.1$ to $0.5$. Since we are not interested in the context acquisition error it is set to $0.0$ in all simulations. The results of this simulation are illustrated in Table 1. The table shows the ratio of high-level prediction accuracy to low-level prediction accuracy ($\frac{P_{high-level}}{P_{low-level}}$). We observe that with a context interpretation error of $0.1$ the high-level prediction accuracy is higher than the low-level prediction accuracy. However, with a context interpretation error of $0.2$ or higher the low-level prediction method achieves the better accuracy.

## V. Conclusion

We have introduced an architecture for context prediction based on low-level context information. Since low-level contexts contain additional information not available with high-level contexts and due to an increased accuracy, common prediction methods are likely to be improved in terms of precision when applied to low-level contexts. Furthermore, additional mathematical operators are available for low-level context data so that novel prediction algorithms may be applied.

We have suggested a context prediction approach based on local alignment methods. This prediction technique utilises the improved knowledge on inter context correlations gained by low-level context prediction. The proposed method is so robust it can handle alterations and even missing or exchanged time series elements. The algorithm incorporates a constant learning mechanism and is able to predict an arbitrary number of future contexts.

First simulation results indicating the potential of the proposed prediction method have been presented as well.

## References


[1] A. K. Dey, "Providing architectural support for building context–aware applications," Ph.D. dissertation, georgia Institute of Technology, November 2000.

[2] M. S. Magnusson, "Repeated patterns in behavior and other biological phenomena," in *Evolution of Communication Systems: A Comparative Approach*, K. Oller and U. Griebel, Eds. Cambridge, MA: MIT Press, 2004, pp. 111–128. [Online]. Available: http://www.isrl.uiuc.edu/ amag/langev/paper/magnusson04inbook.html

[3] G. K. Jonsson, S. H. Bjarkadottir, B. Gislason, A. Borrie, and M. S. Magnusson, "Detection of real-time patterns in sports: interactions in football," in *L'ethologie applique aujourd'hui*, vol. 3. C. Baudoin, 2003.

[4] I. Krsul, "Authorship analysis: Identifying the author of a program," Ph.D. dissertation, Department of Computer Sciences, Purdue University, 1994.

[5] M. S. Magnusson, "Understanding social interaction: Discovering hidden structure with model and algorithms," in *The Hidden Structure of Interaction: From Neurons to Culture Patterns*. L.Anolli, S.Duncan Jr., M.S.Magnusson and G.Riva, 2005.

[6] K. Laasonen, M. Raento, and H. Toivonen, "Adaptive on–device location recognition," ser. LNCS, no. 3001, 2004, pp. 287–304.

[7] D. Ashbrook and T. Starner, "Learning significant locations and predicting user movement with gps," 2002.

[8] B. D. Davison and H. Hirsh, "Predicting sequences of user actions," in *AAAI/ICML Workshop on Predicting the Future: AI Approaches to Time–Series Analysis*, 1998.

[9] P. J. Brown and G. J. F. Jones, "Exploiting contextual change in context–aware retrieval," in *Proceedings of the 2002 ACM Symposium on Applied Computing*, 2002, pp. 650–656.

[10] R. M. Mayrhofer, "An architecture for context prediction," Ph.D. dissertation, Johannes Kepeler University of Linz, Altenbergstrasse 69, 4040 Linz, Austria, Oktober 2004.

[11] L. Barkhuus, "How to define the communication situation: Context measures in present mobile telephony," in *Context*. Stanford, CA: Springer, 2003.

[12] S. Greenberg, "Context as a dynamic construct," in *Human-Computer Interaction*, vol. 16 (2-4). awrence Erlbaum Associates Inc., 2001, pp. 257–268.

[13] P. Nurmi, M. Martin, and J. A. Flanagan, "Enabling proactiveness through context prediction," in *CAPS 2005, Workshop on Context Awareness for Proactive Systems*, June 2005.

[14] R. M. Mayrhofer, H. Radi, and A. Ferscha, "Recognizing and predicting context by learning from user behavior," in *The International Conference On Advances in Mobile Multimedia (MoMM2003)*, vol. 171, September 2003, pp. 25–35.

[15] H.-J. Boeckenhauer and D. Bongartz, *Algorithmische Grundlagen der Bioinformatik*. Teubner, 2003, (in german).

[16] T. Loeffler, S. Sigg, S. Haseloff, and K. David, "The quick step to foxtrot," in *Proceedings of the Second Workshop on Context Awareness for Proactive Systems (CAPS 2006)*, K. David, O. Droegehorn, and S. Haseloff, Eds. Kassel university press, June 12-13 2006.