SesaME: A Framework for Personalised and Adaptive Speech Interfaces

Botond Pakucs
CTT (Centre for Speech Technology)
KTH (Royal Institute of Technology)
Drottning Kristinas väg 31
SE-100 44, Stockholm, Sweden
botte@speech.kth.se

Abstract

This paper presents some motivations for using highly personalised speech interfaces. In particular, the focus is on the requirements for adaptation in mobile environments. Furthermore, SesaME, a framework for personalised and adaptive speech interfaces is described. SesaME supports a multi-domain approach and event-based, asynchronous dialogue management. Finally, a description of how SesaME is employed within the framework of the PER demonstrator is given.

1 Introduction

In natural spoken human-to-human communication, speakers tend to adapt to each other and to the communicative situation. Adaptation is therefore a desirable feature for making speech interfaces more natural. Furthermore, it is commonly believed that spoken dialogue system performance could be enhanced through adaptation.

In this paper, some challenging issues related to adaptation in dialogue systems are discussed. The focus is in particular on the requirements for adaptation in mobile and ubiquitous environments (Section 3). Based on these challenging issues, motivations are presented for using highly personalised speech interfaces for adapting and fine-tuning the interaction to individual users and their actual situation (Section 4) .

The paper also introduces SesaME, a framework for personalised and adaptive speech interfaces (Section 5). SesaME is designed to support a multitude of highly distributed applications and to adapt to individual users and their environment. SesaME features an event-based, asynchronous dialogue management and supports a dynamic plug-and-play solution which enables a multi-domain approach. Furthermore, a solution is proposed to achieve context-based adaptation to an individual user. Finally, the use of SesaME within the framework of the PER demonstrator is described.

2 Background

In the context of dialogue systems, adaptation is the modification of the system's functionality according to the changing circumstances and to the variations in the system input. Based on the source of the variations two major categories of variation can be distinguished. Variations caused by the changing situational circumstances are the *context-related variations*. The *user-related variations* can be attributed to various differences in user characteristics, behaviour and preferences.

It has to be emphasised that there is no clear boundary between the variations attributed to the user and the variations caused by the changing circumstances in the context. Variations in the context may affect the user's behaviour and induce different user related variations. For instance, time pressure and increased cognitive load on the user may cause variations in the user's speech (Müller et al., 2001).

As the user characteristics and the parameters of the situation may change during the same interaction, both the characteristics of the context and the user's properties have to be considered simultaneously. The usability of a system can be expected to increase if both the user properties and the user's situation are taken into account (Jameson, 2001). Thus, *simultaneous adaptation to the user and to the context* is a major challenge for providing more natural and conversational speech interfaces.

2.1 Adaptation to the User

From a practical point of view, not all kinds of user related variations are equally interesting for a dialogue system, see Figure 1. Only a subset of *all theoretically possible feature variations* are actually realised in practice. This is the set of *practically possible variations.* For a realistic dialogue system, which encounters just a limited set of users and handles a limited domain, only a subset of these practically realised variations are usually relevant. These are the *interpersonal variations.*

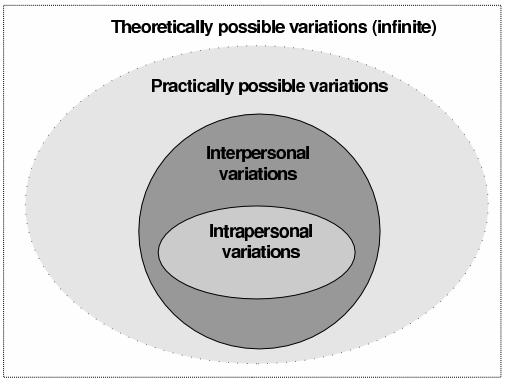

Figure 1: User feature variations.

A robust dialogue system should be able to handle all of the possible interpersonal feature variations relevant to the domain. No single user, however, realises all these possible interpersonal variations. The variations produced by a single user are the *intrapersonal variations.* These intrapersonal variations carry information which may be significant for the interaction with the specific user.

While a speech interface should support interaction with all potential users, the interaction should also be adapted and fine-tuned to the current individual user. During a single interaction a dialogue system has to be able to identify and capture the intrapersonal variations and distinguish them from the interpersonal variations. Consequently, a key issue is how to *model generic user characteristics and specific individual user features simultaneously* (Jameson and Wittig, 2001).

User modelling has been successfully employed in several different dialogue systems (Zukerman and Litman, 2001). However, how to model and exploit data about users in general and data about the current individual user still remains a central challenging issue.

2.2 Adaptation to the Context

Situational[1] factors are used in human-to-human communication to facilitate a smoother and more natural interaction. Context-of-use is relevant knowledge about the actual situation, conversational partners, domain, topic etc. The particular situation is, however, not entirely defined by external variables, but depends also on how users conceptualise the situation (Fischer, 2000). Thus, context-of-use also includes the user's physical and perceptual, social and emotional informational state (Bunt, 1994).

Most experimental and commercial dialogue systems are single-domain dialogue systems. Therefore, with regard to the physical and perceptual context a more or less static situation is usually assumed. However, the growing interest for using speech interfaces in mobile and ubiquitous computing environments makes it necessary to develop better solutions for *capturing, modelling and adapting to the variations in the users' physical and perceptual context.*

Speech-based interaction with ubiquitous services in mobile environments differs from interacting with desktop or telephony-based speech interfaces. Being on the run, and with hands, eyes and sometimes even the mind busy, the users' requirements on the speech interfaces can be ex-

[1]Some researchers make distinction between the notions of "context" and "situation". In this paper, these terms are used merely as synonyms.

pected to increase. Accordingly, in mobile environments it is even more important to fine-tune the speech based communicative interaction to the specific user and to the user's current situation. Some of the *requirements on speech interfaces in mobile and ubiquitous environments* are discussed next.

3 Requirements in Mobile Environments

In dynamically changing mobile environments, the user's intentions and needs may change rapidly. The user should be able to initiate a new task while waiting for some other specific service to be completed. The user should have the possibility to easily cancel a previously issued command or change the parameters of some previously initiated service. Furthermore, the system itself should be able to interrupt an ongoing dialogue and direct the user's attention to higher priority events taking place in the user's immediate environment (physical or virtual).

Thus, in mobile environments it is desirable to support a wide range of domains within one and the same dialogue. Thus, a *multi-domain approach* (Chung et al., 1999) is necessary to allow the user to transparently and seamlessly switch between several topic domains and services. Similarly, *asynchronous dialogue management* (Boye et al., 2000) should also be supported by speech interfaces in mobile environments.

Employing speech interfaces in mobile environments may even introduce some new user requirements. For instance, *means to coordinate and control multiple concurrent speech interfaces* may become necessary in order to avoid the introduction of new usability related problems. When several services and appliances, embedded in the same environment, are listening for user commands, or even taking initiative pro-actively, it is possible, due to errors or misrecognitions, that several speech-based services may be triggered by a single user utterance. To our knowledge, the effects of interacting with several concurrent speech interfaces at the same time have never been studied.

In mobile environments, the users should be able to concentrate upon the task to be performed and not be forced to cope with interface issues. The *usability and user interface issues* should be considered for whole environments rather than for isolated services and appliances. Consequently, support for adaptation, situation awareness and user modelling appears to play a central role for providing natural, user-friendly conversational speech interfaces in mobile environments.

4 Personalised Speech Interfaces

Employing personalised speech interfaces appears to be a promising solution for adapting and fine-tuning the interaction to individual users and their actual situation. Handling some of the above presented challenging issues could also be facilitated by the use of personalised speech interfaces.

As the identity of the user is assumed to be known, there is no need to detect the category or the identity of the user. Only the user's context-of-use has to be detected and modelled at runtime. Consequently, user modelling and adaptation to the user can be regarded as a long term adaptation while adaptation to the context can be regarded as short term adaptation. Thus, *simultaneous adaptation to both the context and to the user* is facilitated.

While using a personalised solution, only the characteristics of a single individual user have to be modelled. Thus, the *simultaneous modelling of generic user characteristics and specific user features* is also facilitated.

4.1 The human-centered approach

The *human-centered approach* is a feasible solution, proposed for achieving personalisation of speech interfaces which are intended to be used in mobile environments (Pakucs, 2003).

According to this approach each user is expected to use an individual and *highly personalised universal speech interface* to access a multitude of services and appliances. Access to the locally available services and appliances is handled through the personalised speech interface, integrated into some personal and wearable appliance such as a mobile phone or a PDA. Application-specific data, including dialogue management capabilities, domain knowledge etc., has to be encoded in *service descriptions* and stored locally at the service provider side. Whenever the user

enters a new environment, the available, distributed service descriptions, have to be dynamically loaded into the personalised speech interface.

The human-centered approach provides extensive support for adapting and fine-tuning the interaction to individual users. For instance, it is even feasible to use speaker-dependent automatic speech recognition which may reduce the amount of the speech recognition errors. Due to the distributed functionality of the human-centred approach, all local services and appliances could become available for all potential users in spite of dialects or non-native accents.

Generally, the requirements on the amount of data necessary for machine learning employed in user modelling and adaptation are considerably high. While using a highly personalised solution, the same speech interface is used for accessing a multitude of services. Thus, *more data becomes available* for machine learning. Consequently, user modelling and adaptation is facilitated.

A highly personalised universal speech interface appears to be ideal for coordinating and controlling *multiple concurrent speech interfaces* and for supporting a *multi-domain approach*. By employing a personalised speech interface, the handling of the potential *security and integrity issues* are also facilitated. Furthermore, personalised speech interfaces are expected to provide an unobtrusive, *user-friendly interaction* and seamless access to services and appliances in mobile environments (Pakucs, 2003).

5 The SesaME Framework

SesaME, shown in Figure 2, is a generic, task-oriented dialogue manager specially designed for the human-centered approach as well as for mobile environments.

SesaME features a blackboard and agent based architecture. The central blackboard stores the representation of the *information state* (Larsson and Traum, 2000) of the dialogue. However, this representation is not formalised; the information state is merely a collection of all data available to the dialogue system. The update of the information state is event-based, where events can be dialogue moves, internal events, or changes in the user's external context. The event-based functionality enables an asynchronous information processing (Blaylock et al., 2002). The SesaME architecture and the theoretical considerations behind it are in some aspects comparable to other agent-based architectures such as the Jaspis (Turunen and Hakulinen, 2000) and the TRIPS (Allen et al., 2000) architectures.

SesaME relies on the ATLAS generic speech technology platform (Melin, 2001). The ATLAS platform provides high-level primitives for basic speech I/O, but access to low-level data is also facilitated.

5.1 A Multi-Domain Approach

One of the key-issues in the SesaME architecture is to support a multi-domain approach. The locally available service descriptions, including dialogue descriptions, recognition grammars and domain knowledge has to be dynamically loaded and activated on the fly. For handling these requirements, a dynamic plug-and-play functionality of the dialogue management capabilities has been developed (Pakucs, 2002).

In SesaME, most of the operations related to the plug-and-play functionality are carried out by the *Dialogue Engine* (DE). Synchronisation and communication with the mobile service environment is taken care by the *Application Interface*. Whenever new changes occur in the service environment (e.g. new services become available or existing services disappear etc.), the Application Interface dynamically updates the *Dialogue Description Collection* (DDC). All currently available dialogue descriptions are stored in the DDC. Beside the different task- and domain-specific dialogue descriptions, DDC also contains resident application independent dialogue descriptions used for error handling or for meta-dialogues necessary for providing information on available services.

For interoperability reasons, the application-specific data, including the dialogue descriptions has to be described in some standardised way. VoiceXML[2] was chosen as the dialogue description formalism. The main reason is the fact that VoiceXML is a markup language which

[2]Voice Extensible Markup Language. For more info see: http://www.w3.org/TR/voicexml20/

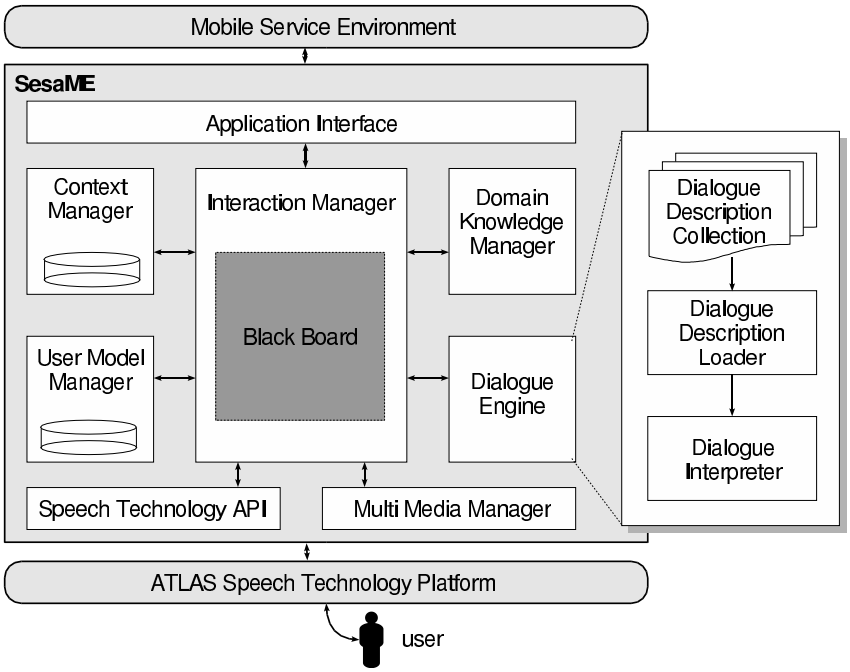

Figure 2: SesaME system architecture with the Dialogue Engine.

shields the developers from low-level implementation details, facilitating rapid application development. VoiceXML also provides support for simple grammars. However, the VoiceXML formalism was slightly extended for allowing additional system prompts used for adaptations and keywords used for topic detection.

Procedurally, the internal plug and play functionality can be divided into three main parts:

- *identification of the task & topic* and the associated dialogue description,
- *activation* of the identified dialogue description,
- the actual *dialogue management.*

At the current stage, the identification of the correct dialogue description is based on topic vectors and keywords extracted from the VoiceXML documents. However, context models, user models or plan based mechanisms could also be used for this purpose.

During the activation, the appropriate dialogue description is translated into internal data structures appropriate for the dialogue management in SesaME. This process is performed through JAXB[3] which provides a fast way to create a two-way mapping between XML documents and Java objects (Pakucs, 2002).

5.2 Dialogue Management

During the dialogue management, the generated internal data structures are used for frame-based dialogue management. The actual dialogue management process does not follow the VoiceXML specifications. Accordingly, VoiceXML is only used as a domain description language.

During the dialogue activation process, two additional parallel data structures are generated. These data structures can be accessed by the *Context and the User Model Managers* and by the *Domain Knowledge Manager*. These data structures may be updated with suggested pieces of information, which can be used to adapt the interaction to the situation and to the user.

Generic, application independent features of the dialogue management are handled by the *Interaction Manager* (IM). The IM also handles er-

[3]The Java Architecture for XML Binding, available at: http://java.sun.com/xml/jaxb/

ror detection, planning, keeps track of dialogue history, and coordinates the different system components and knowledge sources. A central, shared information storage, a blackboard, and a collection of autonomous software agents are the main components of the IM. The detailed description of the IM's functionality is, however, beyond the scope of this paper.

5.3 Adaptation to the Context

A major goal in SesaME is to make full use of the potentials of the individual user models and to achieve context-based adaptation. The adopted solution is inspired by attempts to achieve context-aware computing in the research field of ubiquitous computing.

According to a proposed solution (Dey, 2001) the different possible context types are categorised in *primary* and *secondary* context types. The primary context types characterising the situation of a particular entity consist of the *location* and the *identity* of the entity, the *time* of the interaction, and the *activity* being performed. The secondary pieces of context share a major common characteristic: they can be indexed by the primary context because they are attributes of the entity with primary context. For example, a person's phone number is a secondary type of context and can be obtained through the primary context.

In SesaME, after an interaction with the user every utterance is represented as a feature vector containing feature-value pairs of all relevant information (such as topic, start time of the utterance, length of the utterance, user choices etc.). The only common property of the features in the feature vector is the co-occurrence. The feature vectors are indexed and stored in the user model.

The user model is represented as an inverted file, a common data structure used in information retrieval applications. For manipulating the user model, common information retrieval solutions are used. Accordingly, the user model is domain and task independent and is automatically built. The user model is not formalised in some specific knowledge-based structure. However, it is still possible to apply machine learning solutions such as memory-based learning or similarity-based reasoning.

The *Context Manager* keeps track of the current context. During a new interaction, based on available contextual information, similar interactions are retrieved from the user model. These retrieval results can be used to predict specific features of the ongoing interaction and to achieve adaptation to the current context.

For example, based on earlier interactions with a voice controlled elevator it may be possible to detect that the user's most frequent choice of semantic object was the "fifth floor" when answering to the standardised prompt: *"Which floor would you prefer?"*. Thus, it is possible to predict that the user may want to take the elevator to the fifth floor. By using the additional prompt supported in SesaME, it is possible to ask the user a more natural question: *"Fifth floor, as usual?"* instead of the impersonal standardised prompt.

For enabling the use of alternative questions, the additional system prompts are used:

```
<prompt>
Which floor would you prefer?  </prompt>
<alt-prompt>
<value expr=''predicted-floor''/> floor
as usual?  </alt-prompt>
```

If there are no similar interactions, or no obvious patterns are present in the previous interactions (such as a CD purchasing task), then the default standardised prompt is used.

6 Application

Before evaluating SesaME as a generic, adaptive and personalised dialogue manager in mobile and multi-domain environments, it is necessary to evaluate it as a traditional domain-dependent dialogue manager. In the first application of SesaME, the focus is on the evaluation of the plug-and-play functionality. This evaluation is conducted within the framework of the PER (Prototype Entrance Receptionist) project (Pakucs and Melin, 2001).

6.1 The PER demonstrator

PER, Figure 3, is an animated-agent based automated receptionist located at the entrance of our department. Originally, the system was developed to allow fast and robust access for employees. The application's functionality is stream-lined for this

purpose. PER features a multilingual speaker verification system for Swedish and English. An employee, when approaching the gate, is expected to say his/her password, which consists of the employee's name and a random digit sequence displayed on the screen.

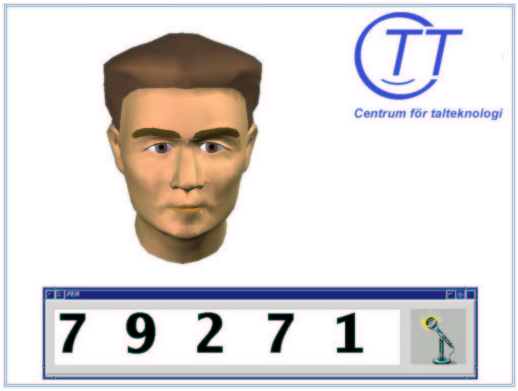

Figure 3: A screen shot of the PER demonstrator.

6.2 Employing SesaME

The functionality of PER has been extended to allow handling of external visitors as well. The interaction with the visitors relies on the SesaME dialogue manager and makes use of VoiceXML-based dialogue descriptions. PER features several different dialogue descriptions associated with different types of visitors and visitor goals such as expected personal guests, seminar visitors, or students attending lectures. The handling of these dialogue descriptions, the topic-based identification of the correct dialogue description, the activation of the identified dialogue description and the dialogue management, is performed according to the description provided in the previous section.

The multi-domain approach allows the incremental updating of the system with new tasks even if the potential of dynamically updating the DDC with new dialogue descriptions at run-time is not fully employed. We plan, for instance, to add dialogue descriptions to help to guide the visitors in the building. The plug and play functionality allows extension of the application for support of new languages. Adding support for English speaking visitors is also planned in the near future.

The domain-dependent data necessary for the dialogues is stored in an external database and is made available for manipulation through the web. Thus, employees can easily add to the database expected visitors, detailed information on seminars or lectures, or they can change the parameters of existing entries. In this way the data upon which PER operates and upon which the dialogue descriptions are generated is always kept up to date.

7 Conclusions

This paper introduced SesaME, a framework for personalised and adaptive speech interfaces. SesaME features a blackboard and agent based architecture which supports event-based, asynchronous dialogue management. The employed dynamic plug-and-play dialogue management solution enables a multi-domain approach and the run-time accessing of the locally available services in mobile and ubiquitous environments.

Some parts of the SesaME architecture are still under development. However, it has already been successfully employed in the framework of the PER demonstrator. The application demonstrates that it is feasible to support a multi-domain approach through a dynamic plug-and-play solution while still allowing a generic and flexible dialogue management.

A major goal of the SesaME framework is to support highly personalised speech interfaces and to facilitate the adaptation and fine-tuning of the interaction to individual users and their actual situation. A description of the solution employed in SesaME for achieving context-based adaptation to individual users' current situation was also described.

Using highly personalised speech interfaces appears particularly advantageous in mobile and ubiquitous computing environments. The suggested framework creates novel possibilities for supporting personalisation, context awareness and user modelling in dialogue management. The adaptation and usability related advantages are interesting enough to make the proposed framework worthy of further development and evaluation.

8 Acknowledgements

This research was carried out at the CTT, Centre for Speech Technology, a competence centre at KTH, supported by VINNOVA (The Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organisations. This work was also sponsored by GSLT, the Swedish National Graduate School of Language Technology and by the European Union's Information Society Technologies Programme under contract IST-2000-29452, DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces).

References

James Allen, Donna Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2000. An architecture for a generic dialogue shell. *Natural Language Engineering*, 6(3):1–16, December.

Nate Blaylock, James Allen, and George Ferguson. 2002. Synchronization in an asynchronous agent-based architecture for dialogue systems. In *Proceedings of 3rd SIGdial Workshop on Discourse and Dialogue*, June.

Johan Boye, Beth Ann Hockey, and Manny Rayner. 2000. Asynchronous dialogue management: Two case studies. In David Traum and Massimo Poesio, editors, *Proceedings of GÖTALOG 2000*, Gothenburg, Sweden, June.

Harry Bunt. 1994. Context and dialogue control. *THINK Quarterly*, 3(1):19–31.

Grace Chung, Stephanie Seneff, and Lee Hetherington. 1999. Towards multi-domain speech understanding using a two-stage recognizer. In *Proceedings of Eurospeech '99*, pages 2655–2658, Budapest, Hungary, September.

Anind K. Dey. 2001. Understanding and using context. *Personal and Ubiquituos Computing*, 5(1). Special issue on Situated Interaction and Ubiquitous Computing.

Kerstin Fischer. 2000. What is situation? In *Proceedings of GÖTALOG 2000*, Gothenburg, Sweden, June.

Anthony Jameson and Frank Wittig. 2001. Leveraging data about users in general in the learning of individual user models. In B. Nebel, editor, *Proceedings of IJCAI 2001*, pages 1185–1192, San Francisco, CA, USA. Morgan Kaufmann.

Anthony Jameson. 2001. Modelling both the context and the user. *Personal and Ubiquitous Computing*, 5:29–33.

Staffan Larsson and David R. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340, September.

Håkan Melin. 2001. ATLAS: A generic software platform for speech technology based applications. *TMH-QPRS,Quarterly Progress and Status Report*, 42.

Christian Müller, Barbara Großmann-Hutter, Anthony Jameson, Ralf Rummer, and Frank Wittig. 2001. Recognizing time pressure and cognitive load on the basis of speech: An experimental study. In *UM2001, User Modeling: Proceedings of the Eighth International Conference*.

Botond Pakucs and Håkan Melin. 2001. PER: A speech based automated entrance receptionist. *Presented at the 13th Nordic Computational Linguistic Conference, NoDaLiDa 2001*, May. Available at: http://www.speech.kth.se/~botte/.

Botond Pakucs. 2002. VoiceXML-based dynamic plug and play dialogue management for mobile environments. In *Proceedings of ISCA T&R Workshop on Multi-Modal Dialogue in Mobile Environments*, Kloster Irsee, Germany, June.

Botond Pakucs. 2003. A human-centered approach to speech interfaces in mobile and ubiquitous computing environments. *TMH-QPRS,Quarterly Progress and Status Report*, 45. Available at: http://www.speech.kth.se/~botte/.

Markku Turunen and Jaakko Hakulinen. 2000. Jaspis - a framework for multilingual adaptive speech applications. In *Proceedings of 6th International Conference of Spoken Language Processing (ICSLP 2000)*, Peking, China.

Ingrid Zukerman and Diane Litman. 2001. Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, 11:129–158.