Multiple Workspaces as an Architecture for Cognition

Jeremy L Wyatt and Nick Hawes

Intelligent Robotics Laboratory
School of Computer Science
University of Birmingham
Birmingham, UK, B15 2TT
{jlw,nah}@cs.bham.ac.uk

Abstract

In this paper we describe insights for theories of natural intelligence that arise from recent advances in architectures for robot intelligence. In particular we advocate a sketch theory for the study of both natural and artificial intelligence that consists of a set of constraints on architectures. The sketch includes the use of multiple shared workspaces, parallel asynchronous refinement of shared representations, statistical integration of evidence within and across modalities, massively parallel prediction and content addressable memory to allow binding across workspaces.

Introduction

In this workshop we are considering the influence of evidence from biology on architectures for cognition. In this paper we suggest that there is much to be gained from ideas flowing in the opposite direction; that past and current ideas from engineering intelligent robots have much to teach us about cognition in nature. A weaker statement is that, at the very least, computational theories of sensing and action from robotics and biology are currently converging around a number of simple but powerful ideas. In particular we suggest a simple combination of ideas as a theory of cognition that is worth investigation by other researchers.

The paper should be read with several caveats. First, we do not propose this theory as being either complete or correct. It provides a starting point to be refined. We expect parts of what follows to be dropped or augmented in future revisions: this is our best first guess. The second important point is that we are concerned with agents that have bodies. Third, we are interested in describing a space of possible architectures, by enumerating constraints on those spaces. This follows the approach of Sloman [13]. Fourth, when we talk about a theory of cognition we want to be clear that we are concerned with these ideas as theories of both artificial and biological cognition (with an emphasis on human cognition) at quite a high level of abstraction. In other words we are describing virtual machines that may be implemented in a variety of different ways at the hardware level. We call our sketch theory a Multiple Workspace Theory of cognition (MWT). The basic components of this theory are:

1. Multiple shared workspaces: Processing of information in a cognitive system can be grouped effectively around a number of shared workspaces.
2. Parallel refinement of shared representations: Processing components that transform information do so in parallel, and the hypotheses they produce are posted and mutually refined in these shared workspaces.
3. Statistical integration of evidence: Mutual refinement of these hypotheses is performed in part by Bayesian reasoning using asynchronous message passing.
4. Many simple predictors: Prediction is performed by merging the predictions of many predictors that are context specific experts, rather than by few generalist predictors.
5. Content based memory: Creation of sensory expectation can occur by projection across workspaces via the conditional probabilities that make associative memory.

These ideas allow us to pose a number of additional hypotheses, some of which are directly testable:

1. The final fusion of action is performed through a single shared workspace.
2. The theory poses four problems that any effective cognitive system specified within it must solve: the binding problem, the filtering problem, the action fusion problem, and the processing management problem.
3. There may be a number of privileged workspaces that receive and transmit information globally, whereas other workspaces receive and transmit information locally.
4. Workspaces with a global sensory-motor footprint tend to manipulate information at a more abstract and a-modal level than workspaces with a local footprint.

We now describe how the theory draws on ideas from work on cognitive architectures and robotic architectures. We discuss requirements for robotic systems that have driven our previous work on architectures for robotic cognition. We then proceed to sketch the MW theory.

Ideas from research on cognitive architectures

There have been several attempts at unified theories of intelligence from within cognitive science. At least two of these emphasise the role of production systems. In SOAR Newell and Laird [9] proposed a production system model in which serial application of rules, written in a common form, modified representations held in a workspace shared by those rules. An important component of the theory was that there was a single unified representational language within which all data held in the shared workspace was expressed. Another key idea was that a set of meta-rules controlled the serial application of these productions. These three key ideas: a single shared workspace, serial application of processing elements, and a common representational language have been extremely influential. They are both simple to comprehend and allow the construction of effective systems.

In ACT-R [1] John Andersen and colleagues have taken some of the elements of production systems and used them to produce models of aspects of human cognition that produce testable predictions. In ACT-R productions now represent the serial actions of processing in the thalamus and connect to information in buffers. Together these simulate the behaviour of multiple thalamic-cortical loops. ACT-R has been used to construct models that give impressively accurate predictions for human performance on a range of tasks, including reading and mental arithmetic. ACT-R models rely heavily on the provision of timing information about delays in each stage of processing. Both SOAR and ACT-R have in common the fact that they have widely available languages that allow researchers to implement computational models.

Finally in Global Workspace Theory (GWT), Baars and Shanahan [12] have proposed the idea that conscious thought is explainable at an abstract level by the idea of a global workspace. The key idea in GWT is that local processes propose items to be posted onto a single global workspace, and that mechanisms exist that select one collection of items that are in turn re-transmitted to all the local processes.

It can be seen that all three theories emphasise the idea of a single shared workspace for parts of cognition. There is evidence however, at least from robotics, that such a single workspace is an incomplete architectural account of intelligence. We now turn to describe ideas from robotics on architectures, in order to compare and contrast them with the ideas from work on cognitive architectures.

Ideas from research on robotic architectures

The first significant attempt to implement what might be loosely called a cognitive robot was the Shakey project [10]. Detailed examination of their approach bears fruit. The architecture in Shakey was dominated by a central workspace within which all data about the contingent state of the world was written in a single representational language. In the case of Shakey this was a form of first order predicate logic minus quantification. Sensory processing was essentially a business of abstracting from the raw sensor data to this predicate description. Typing of entities was captured using predicates, and the representation also captured some metric information for the highly simplified world. Qualitative action effects were captured using STRIPS operators, which addressed some of the difficulties previously encountered by the situation calculus. This declarative knowledge about action effects was used in a planning process that had available all knowledge in the world model. In addition to this the robot had fixed routines that would update the world model when sufficient uncertainty had accumulated about its state. This was the way that gross error recovery occurred: through re-sensing and then recalculating the world model. Simpler errors were handled using Intermediate Level Actions (ILAs). These were essentially discrete closed loop controllers that relied upon the world to settle between each step and thus couldn't deal easily with ongoing rapid change. Overall, Shakey shares, at an architectural level, some of the assumptions of SOAR and ACT-R. It relies on serial application of planning operators to simulate trajectories through the state space and selects courses of action based on those. It uses a single representational language, although it does reason with that representation using two different kinds of reasoning. It has completely serial control of execution: only one ILA is in control of the robot at a time. It collects all contingent knowledge about the world in a single shared workspace. It handles error recovery in two ways: re-sensing leading to model updating and re-planning, and closed loop recovery from errors without planning. Finally it does not provide an architectural answer to the problem of sensory interpretation: perceptual routines existed in Shakey, but there are no architectural constraints or aids to how they operate, communicate, or share information. They serve only to provide information to a central model in a unified language.

The classic story about Shakey given by behaviour based roboticists, is that it could never have worked outside of its carefully controlled environment, and that even within it performance was unreliable. Shakey was able to perform different tasks, but it relied upon an accurate world model. It was able to construct a sufficiently accurate representation under benign sensory conditions, but robot vision researchers unsuccessfully spent the decade following the Shakey project trying to extend this approach of scene reconstruction to more natural visual environments. Thirty-five years later our ability to perform scene or surface reconstruction is still poor, although it has improved. Of course to be fair to the designers of Shakey it is not clear that they took a principled stance on whether all aspects of a scene should be recovered, only that attempts to extend their approach often sought to do this, and have largely failed to date.

A strong reaction to this paradigm that occurred in the mid 1980s was exemplified by the behaviour based approach to robotics [2,3]. The approach is characterised by a number of authors including proponents and sceptics. One key idea is that the system is almost entirely representation free. The meaning of this statement depends on what is meant by representation. Kirsch [7] describes three types of representation: data items the values of which co-vary with features in the world; declarative statements which release their information when queried; and predicate descriptions that allow generalisation by type, leading to the ability to reason about inheritance. Brooks' early robots were certainly representation free on the basis of either of the last two definitions. Furthermore, while modules may have representations of the first kind, they do not typically transmit these representations to other modules for further processing or consumption. In other words there is no sharing of information, only competition for control of the robot. Other important aspects of the approach are that many controllers run in parallel, that each is relatively simple, and that their action recommendations are fused through a single global mechanism.

The obvious weakness of the behaviour based approach is the lack of evidence of its ability to scale to higher cognitive functions, despite nearly a quarter of a century of effort. Instead, roboticists typically use behaviours as the lowest level of control in a hierarchical system. Three tiered architectures such as 3T [4] employ behaviours at the lowest level, and link these to a symbolic planning level via a sequencing level in which transitions are made from behaviour to behaviour using a finite state machine like representation. Representations have made a re-appearance via advances in filtering and statistical approaches imported from machine learning. Behaviours, have thus been merged into the tool-box of techniques employed by most roboticists within architectures, rather than being an architectural choice in their own right.

Requirements for architectures from robotics

In order to understand why a particular architecture or space of architectures has properties suitable to support cognition in creatures with bodies we need to think about the requirements that those systems must satisfy. Roboticists have plenty of experience of these. Our own work has been in building robot systems that are able to interact with humans, either while mobile in an office setting, or while manipulating objects in a space shared with a human. There are several properties of the interactions that must be handled by the architecture:

Dynamism: the world changes rapidly and independently of the robot.
Uncertainty: sensors are inaccurate, and actions fail.
Multiple modalities: the robot must use information from multiple sensory modalities in order to make decisions. In our case these include simple haptics, vision, proprioception, and speech. It also has multiple motor systems, and the actions of these must be coordinated.
Re-taskability: the robot must be re-taskable. It should be goal directed, but must still be able to tailor its behaviour to the context, or switch to a new task if necessary.

These are quite general *run-time* requirements on the interaction, i.e. requirements on the system during performance. Out of these arise run-time and *design-time* requirements on the robot architecture itself, some of which are imposed by the state of the art in engineering. I do not suggest that these are necessarily shared by natural systems, but they have influenced our architectural choices. They include:

Parallel processing: many of the algorithms employed are computationally demanding. A serial model of processing is thus unworkable. There is a need for perceptual components to run in parallel. Action components must also run in parallel so that the robot can do more than one thing at a time, e.g. looking and reaching.
Asynchronous updating: information arrives in different modalities at different rates. In addition processing in some modalities is slower than in others. This requires us to accept that updating of information will occur asynchronously across the system.
Multiple specialist representations: the field of AI has fragmented, and the sub-disciplines have developed their own specialist representations for inference and decision-making. In designing robots with multiple forms of sensing and acting we need to bridge the gaps between them.
Understandability: the systems we build are complex, and current engineering methods rely on most of the tokens in the system being semantically transparent.
Incrementality: it is desirable that our systems are extendable. One of the requirements for this is a kind of modularity that supports extendability. When we add additional sub-systems the degree of re-engineering of other sub-systems should be minimised.

Given these constraints the management of information flow in the robot system becomes key. How should information from one sub-system be communicated to others? How should decisions to act be combined and sequenced? How should we determine whether separate pieces of information are related? These are questions to which we should provide architectural answers. The insights from the requirements above are that aspects of all the architectural solutions described above are required. The strength of the behaviour-based approach is its use of parallelism. In our Multiple Workspace theory we use parallelism, but apply it to representation not just action. The insight from production systems is the idea of a shared workspace. But rather than limit ourselves to a single workspace we suggest that multiple workspaces have some advantages. These include the ability to support parallelism in both perception and action; while still making exchange of representations central to cognition; and managing information by grouping to ameliorate the difficulties of information management in single workspace architectures. These advantages will be discussed further below. Next I sketch a possible Multiple Workspace theory of cognition.

A multiple workspace theory of cognition

In the introduction to the paper we stated five aspects we believe should form part of a theory of cognition. The first two are drawn from our experience of building robot systems, and for which we therefore already have working implementations. The five components of the theory we outlined were as follows: multiple shared workspaces; parallel refinement of shared representations; statistical integration of evidence; use of many parallel, highly context specific predictors; and use of content addressable memory to link across workspaces. We detail each in turn. The first two have working implementations, and we will describe four important problems that they present system designers with before moving on to the last three.

Tested Parts of our theory

Multiple Shared Workspaces. Our multiple workspace architecture (referred to as CAS in our previous papers [6]) is composed of a set of what we call sub-architectures. Each one contains of a group of processing components, which share a single public workspace. Sub-architectures can pass information to one another, and components can also talk to motors and sensors. Each component can read input from and write output to its shared workspace. Components may also have private memory. Components fall into two types: managed and unmanaged. Unmanaged processes are pre-attentive: they run regardless of the processing resources available elsewhere. Managed processes have to ask for permission to run, and share limited computational resources. The resource allocation is handled by a management component that either permits or orders the managed processes to run. The unmanaged processes operate in a completely data driven manner. They post the results of fast processing of high bandwidth data feeds onto the shared memory. Managed processes can run in a data driven (bottom up) fashion or in a top down (goal driven) fashion. Some sub-architectures have privileges to write to the working memories of other sub-architectures.

Parallel Refinement of Shared Representations. To get a detailed idea of what a robot system looks like when implemented using this architecture schema we commend the reader to [6]. The system described therein can manipulate objects on a table-top, understand utterances about the scene by a human, make its own utterances in reply, and execute commands that are concerned with moving the objects so as to alter their spatial relationships. The overall system is composed of about 35 processing components grouped into seven sub-architectures. Within a sub-architecture, objects on the working memory are refined in parallel by many processes. In the visual sub-architecture, for example we have data objects that provide descriptions of objects in a scene. These contain many different fields. Within each sub-architecture working memory, the separate pieces of information stored are bound together in these data structures by locally relevant features, e.g. by spatial information in vision or syntactic structure in language. We do not suggest that this is a biologically plausible solution at the hardware level, but note simply that others have explored solutions to binding in neural architectures [14]. The key idea is that within a sub-architecture many components can refine parts of shared data structures in parallel. They look for data they wish to process, and post the results back onto the working memory, bound into existing data structures. This idea of parallel refinement is central to our approach.

Issues arising from the first two hypotheses. A system that has multiple modes of sensing and acting, limited computational resources, and distributed representations faces four problems. We refer to these as the *binding* problem, the *filtering* problem, the *processing management* problem, and the *action fusion* problem. These problems are faced by naturally occurring intelligent agents as well. A MW theory can provide plausible solutions to these. We define the problems as follows:

1. **The Filtering problem**: when a piece of information is generated in one part of a system how does the system decide where that piece of information should go? Which sub-systems might need to know about it? There are many architectural design decisions that contribute to the management of information flow. In our MWA the problem becomes one of filtering out the information created by one component that another component doesn't need to know about, as efficiently as possible. The disadvantage of single workspace architectures is that they require all information changes to be posted to all components: this is flexible but inefficient At the other extreme, point to point architectures, particular types of information are only ever sent to the same components, there can be no modulation by context. This is efficient but inflexible. Our MWA achieves efficiency by grouping together components that commonly need to exchange information, thus keeping others insulated from most irrelevant change. It achieves flexibility by allowing workspaces to share limited summary information. Thus it is both flexible and efficient.
2. **The Binding problem**: arises when information created by one component may need to be related to another piece of information elsewhere. Our MWA allows binding at several levels. Intra sub-architecture binding uses local features as mentioned above. Inter sub-architecture binding requires long-term cross modal memories to support matching. In our case we use type hierarchies that allow the comparison of information from different sources by type and value. Thus if we speak about the blue ball, an ontology linking vision and language will allow us to bind the utterance to an entity from vision.

3. **The Processing Management problem**: The system has limited computational resources. How does it allocate these in a way that is appropriate to its current task and context? How does the robot decide which bits of information should be processed further, and what processing should be performed? In psychology the former is known as the problem of attention. However, our robots need mechanisms that determine what processing is done too. If I look at a cup to recognise it, I extract different information than when I look at it to pick it up.

4. **The Action Fusion problem**: Given that an agent has multiple motor systems, and that each may have instructions from a higher executive, perhaps contradictory ones, how should the execution of their instructions be coordinated during execution. This is the problem that behaviour based systems solve. In the brain there are several structures that are implicated in this role, including the cerebellum and the basal ganglia. We suggest that at the abstract level of virtual machines action fusion can be modelled as occurring in a single shared workspace.

Untested parts of our theory

So far we have described the two parts of our theory that have been implemented in a software toolkit (CAST) and in working robots. We now turn to the last three elements.

Statistical integration of evidence. Evidence from sensors is unreliable, and in robotics the dominant way of dealing with this problem is to view sensory signals as being drawn from underlying (or hidden) states according to some probability distribution. In other words the world is not directly observable, we merely have hints and clues as to its state. The problem of perception in this framework is to integrate the cues required to perform our current task, and to do so as reliably and efficiently as necessary. The result of the statistical view is that if we consider the possible hidden states of the world as hypotheses and we can statistically model the processes by which these hypothesised states may generate the sensory information, then we can use Bayes rule (or more specifically Bayesian filtering) to infer a distribution over the true hidden state of the world. The key elements in robotics that we need to be able to model are the observation or measurement model Pr(O|H) where H is the hypothesised world state, and O is the sensory observation; and a forward model, which captures how the state of the world changes under action. With the availability of these two models --- usually via statistical machine learning methods --- and the efficient application of Bayes' rule, robotics has taken significant strides forward in the past decade: Kalman filters, particle filters, the simultaneous localisation and mapping problem (SLAM), the visual SLAM problem. All of these have a Bayesian filtering formulation, or are solutions to such a problem. It is therefore interesting to note that neuroscientists have found strong evidence that some sensory-motor systems in the human brain appear to be integrating evidence according to Bayes' rule [8].

Such a statistical view of perception fits neatly with the idea of a multiple workspace theory of cognition. Shared working memories store the current hypotheses, and different components are used to integrate sensory evidence. Let's imagine trying to estimate the shape of an object in a visual scene. There are cues from edge features, shading, texture distortions and stereo. These are our observations, and represent the consequences of processing by our unmanaged components. We also have hidden variables that capture the aspects of the underlying state we are trying to estimate. If we code the curvature, orientation and depth of each visible surface point as separate variables we can consider that each can be modelled for each patch by a separate variable on the working memory. In addition the variables representing different quantities with respect to the same patch can be linked, and the variables representing neighbouring patches can be linked. In both cases this will allow joint inference since we effectively now have a Bayesian network. If we have a generating model from each hidden variable to each feature cue we can then revise the distribution over the values for each of the hidden variables (or hypotheses) by Bayes' rule whenever a new feature cue arrives on the working memory. In this way we refine our shared hypotheses in a statistically well-founded manner. The advantages of this architecture for perception are manifold. First we have the ability to separate the independent sensory effects of a state of the world. Second we have a natural way to distribute the processing in a neural fashion by creating cells that code for specific values of hidden variables, and also cells for the different feature cues. Finally we have a way of propagating information across the hidden variable cells as we alter the spatial reference: content addressable memory. The connections between entities on the shared memory need to be coded explicitly in long-term memory.

Content Based Memory. How should these connections be coded, and how can we use them? Two-way connections between feature cue variables and hidden variable variables allow us to code the conditional likelihoods of observations given world state and vice-versa. These allow us to create sensory expectations in one modality given information from another, or to do so across sub-architectures. The handling of early binding can be performed by the creation of expectations via long-term content-based memory. This represents, for example, the connection between hidden variable values across sub-architectures. A binding sub-architecture can be used to create links between these entities, in order that hidden variables' values from one sub-architecture (e.g. the presence of blue in the scene) can then be used to revise the likelihoods of hidden variable values from another one (e.g. the likelihood of a part of the speech signal being the word "blue"). This allows us to use long term memory to facilitate early tentative binding, which in turn reinforces inference from sensory information. In our current implementations we do not use probabilistic binding but we do have a separate binding workspace within which entities from vision and natural language utterances are bound together on the basis of long term memory in the form of type hierarchies. The advantage of early binding across modalities in improving the robustness of sensory processing has already been shown in robotics [11].

Multiple Simple Predictors. One of the most important abilities of an intelligent system is the ability to create predictions of the effects of motor actions. Without such an ability we are unable to reason forward to create multi-step plans, or to learn from mental simulation. Since the effects of actions depend heavily on the current state of the world the prediction problem is complex. One solution to this is to employ a small number of very powerful, complex predictors that cover a wide range of contexts. Physics engines are a good example of this. At the other extreme we can employ many simpler predictors that each make predictions for a small number of contexts. If we imagine our predictors as components that can post predictions onto a working memory then we can imagine these predictions as hypotheses. These can be subsequently discarded or retained as evidence accumulates against or for them. Current theories from computational neuroscience include the idea of the sensory motor system being composed of many pairs of context specific predictors and controllers [5]. This idea fits neatly with the idea of a shared workspace. Predictions from many simple context specific experts are posted onto the working memory. Each is also concerned with predicting the outcome for a very small spatial extent in the world. The shared working memory is then used as a space within which these local experts have to refine their predictions, and from which a consistent set of predictions must be selected. In order to understand this idea imagine trying to predict the trajectory of a ball as it rolls towards a wall. The object can have one or two surface contacts. In the case of the ball rolling in free space, one subset of predictors forms a consistent set. In the case where it collides with the wall there are now two contacts, and these cause a different set of predictors to form a consistent set. Using a shared working memory to conduct these resolutions is a natural way to provide this integration of predictions.

Conclusions

Why do these ideas fit particularly well with the idea of a multiple workspace theory? How do they meet the criteria that we set out for robot cognition above? The idea of multiple workspaces with parallel refinement is valuable because it provides an architectural mechanism via which we can deal with reasoning within and across multiple modalities efficiently. Single workspace theories of cognition can give no architectural account of perception and do not scale. Multiple workspaces limit the computational complexity of inference and decision making by corralling the inferential processes so that they do not interfere unnecessarily. Decisions about what is relevant are carried out many times locally and in parallel, rather than being made globally. Combined with Bayesian reasoning we have an architecture which has a principled approach to uncertainty, and which therefore allows us to deal with uncertainty, and with the integration of evidence from multiple modalities in a principled and robust manner. Message passing approaches to Bayesian reasoning are well understood and established, and allow a principled way of dealing with the asynchronous arrival of information. Finally the ability of all workspaces to reason with representations, and of some workspaces which have global footprints to reason at very abstract levels allows us to get away from the lack of true re-taskability of the behaviour based approach. By allowing parallel refinement around shared workspaces we also provide a natural link between the architecture and efficient parallel implementations that can provide real-time performance. For all these reasons we argue that the set of constraints we outline defines the space within which useful theories of biological cognition --- expressed at a level more abstract than that of the architecture of the nervous system --- can be found. Although in this paper we have not had the space to describe in detail our ideas about how content-based memory and the use of many simple predictors fit together with the rest of the ideas we are confident that the ideas we present here will be useful to other researchers.

We gratefully acknowledge the support of the EC grant CogX, FP7-2007-ICT-215181.

References

[1] Anderson, J., Bothell, D., Byrne, M., Douglass, S., Lebiere, C. and Qin, Y. 2004. An integrated theory of the mind. *Psychological Review* 111(4):1036–1060.

[2] Brooks, R., Elephants don't play chess. *RAS* (6):3–15, 1990.

[3] Brooks, R., Intelligence without representation. *Artificial Intelligence*, (47):139–159, 1991.

[4] Gat, E. 1997. On three-layer architectures. In Kortenkamp, D., Bonnasso, R., and Murphy, R., eds., *AI and Mobile Robots*.

[5] Haruno, M., Wolpert, D., and Kawato, M.,. Mosaic model for sensorimotor learning and control. *Neural Comp.*, 13(10):2201–2220, 2001.

[6] Hawes, N., Sloman, A., Wyatt, J., Zillich, M., Jacobsson, H., Kruijff, G., Brenner, M., Berginc, G., and Skocaj, D., Towards an Integrated Robot with Multiple Cognitive Functions. In *Proceedings AAAI '07*, 1548–1553, AAAI Press, 2007.

[7] Kirsch, D., Today the earwig, tomorrow man? *Artificial Intelligence*, (47):161–184, 1991.

[8] Körding, K. and Wolpert, D., Bayesian integration in sensorimotor learning. *Nature* (427):244–247, 2004.

[9] Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. Soar: An architecture for general intelligence. *Art. Intell.* 33(3):1–64.

[10] Nilsson, N. *Shakey The Robot*, Tech Note 323. AI Center, SRI International, Apr 1984.

[11] Roy, D. and Mukherjee, N., Towards Situated Speech Understanding: Visual Context Priming of Language Models. *Comp. Speech and Lang.*, 19(2):227–248, 2001.

[12] Shanahan, M. and Baars, B., Applying Global Workspace Theory to the Frame Problem. *Cognition* 98(2):157–176, 2005.

[13] Sloman, A., Varieties of affect and the Cogaff architecture schema. In *Proc. of AISB'01*, 1–10,2001.

[14] Van der Velde, F., and De Kamps, M., Neural Blackboard Architectures of Combinatorial Structures in Cognition. *Behavioral and Brain Sciences* (29):37–70, 2006.