Artif Intell Rev
DOI 10.1007/s10462-008-9089-y

Embedding intelligent decision making within complex dynamic environments

G. M. P. O'Hare · M. J. O'Grady · R. Tynan · C. Muldoon · H. R. Kolar · A. G. Ruzzelli · D. Diamond · E. Sweeney

Abstract Decision-making is a complex and demanding process often constrained in a number of possibly conflicting dimensions including quality, responsiveness and cost. This paper considers in situ decision making whereby decisions are effected based upon inferences made from both locally sensed data and data aggregated from a sensor network. Such sensing devices that comprise a sensor network are often computationally challenged and present an additional constraint upon the reasoning process. This paper describes a hybrid reasoning approach to deliver in situ decision making which combines stream based

G. M. P. O'Hare (✉) · M. J. O'Grady · R. Tynan · C. Muldoon · A. G. Ruzzelli
The Centre for Sensor Web Technologies, School of Computer Science and Informatics,
University College Dublin, Belfield, Dublin 4, Ireland
e-mail: Gregory.OHare@ucd.ie
URL: http://www.clarity-centre.com

M. J. O'Grady
e-mail: Michael.J.OGrady@ucd.ie

R. Tynan
e-mail: Richard.Tynan@ucd.ie

C. Muldoon
e-mail: Conor.Muldoon@ucd.ie

A. G. Ruzzelli
e-mail: Ruzzelli@ucd.ie

H. R. Kolar
Big Green Innovations, IBM Systems and Technology Group, T.J. Watson Research Centre,
New York, USA
e-mail: kolar@us.ibm.com

D. Diamond
The Centre for Sensor Web Technologies, Adaptive Sensors Group, NDRC,
Dublin City University (DCU), Glasnevin, Dublin 8, Ireland
e-mail: Dermot.Diamond@dcu.ie

E. Sweeney
The Marine Institute, Rinville, Oranmore, Co. Galway, Ireland
e-mail: Eoin.Sweeney@marine.ie

computing with multi-agent system techniques. This approach is illustrated and exercised through an environmental demonstrator project entitled SmartBay which seeks to deliver in situ real time environmental monitoring.

Keywords Intelligent agents · BDI agents · Hybrid intelligence · Distributed decision-making · Resource bounded reasoning · Stream computing · Environmental monitoring

1 Introduction

Decision making is a complex and demanding process often constrained in a number of possibly conflicting dimensions including quality, responsiveness and cost. Decision delivery is governed by a variety of Quality of Service (QoS) parameters, which must be configured according to the particular context within which a decision is required. In certain safety critical circumstances, priority will rest with the overall quality and confidence that may be ascribed to a given decision. In highly dynamic scenarios, latency and responsiveness may be of paramount importance while in embedded in situ applications that are governed by limited computational and power availability, efficiency (cost) of the decision may prove singularly important.

As context varies so the decision making process needs to adapt. This demands intelligent decision-making and often necessitates the use of a number of distinct reasoning approaches. Such hybrid intelligence approaches enable the accommodation of the benefits of complimentary techniques. This paper presents a hybrid reasoning approach that fuses stream based computing with collaborative reasoning as realized through multi-agent system techniques. Often such a decision-making apparatus is embedded within the environment and hosted upon a distributed network of ambient devices. This approach is illustrated and exercised through an environmental demonstrator project entitled SmartBay, which seeks to deliver in situ real time environmental monitoring.

2 Related research

Intelligent systems have been the subject of active research since the 1960s. As a result of this, many techniques have been developed including neural networks, fuzzy reasoning, machine learning, case-based reasoning, decision support systems and intelligent agents. Each technique encompasses characteristics that make it more apt for certain problem domains. However, it has been recognized that a select and prudent combination of AI techniques can result in a potent solution for particular problems, resulting in so-called hybrid intelligent systems. A number of such systems have been described in the literature. These have been deployed in various domains including terrain mapping (Fregene et al. 2005), health management (Fleming et al. 2007), urban planning (Wu 1998) amongst many others. For the purposes of this discussion we examine a number of exemplary systems that focus on issues pertaining to the monitoring of various aspects of the environment.

Recknagel et al. (1994) describe an expert system DELAQUA that assists in the control of water quality in lakes and reservoirs. This early example is constructed using expert system technologies and harnesses fuzzy models for simulation purposes. Likewise in the water domain, Taha and Ghosh (1995) have developed a hybrid intelligent system for the control of water reservoirs. Their system incorporates select aspects of rule-based systems and connectionist architectures. Successful wastewater treatment is a priority for sustainable living;

Feng et al. (2003) describe the construction of a virtual sensor for water quality control that incorporates neural networks and fuzzy logic. In maritime environments, Chen and Mynett (2006; Chen et al. 2006) use a fuzzy cellular automata approach for modeling algae blooms in Dutch coastal waters. As can be seen from this brief description, the use of AI and hybrid intelligence has been applied quite a number of years in this singular aspect of environmental informatics. More applications exist and the interested reader is referred to Chen et al. (2006; Chen and Mynett 2006) for an overview of the use of AI for modeling environmental systems, and to Chan and Huang (Chan and Huang 2003) for their use in pollution management.

A major challenge facing AI researchers concerns the deployment on devices of limited computational capacity such as mobile phones, sensors and other embedded computational artefacts. Embedded Agents (O'Hare et al. 2006; O'Grady and O'Hare 2004, 2005; Keegan et al. 2008; Muldoon et al. 2003), for example, and neural networks (Heaton 2005) are two candidate technologies that have been deployed upon mobile devices. However, realizing effective reasoning strategies in such circumstances can prove computationally intractable using traditional approaches. However, new approaches are under investigation and involve consideration of energy implications associated with the reasoning process itself together with the cost of affecting any associated course of action. For a more detailed description of the issues involved, the reader is referred to Shen et al. elsewhere within this special issue. In this paper we focus upon two particular instruments for delivering decision-making within the associated application constraints, these are those of stream processing and collaborative decision making systems which we each consider in turn.

3 Stream processing

In the case of continuous data environments with heterogeneous data types and dissimilar data streaming rates, intelligent stream processing has been of interest in the networking and database domains. The processing of streaming data may be thought of in terms of two components: handling (ingestion, logging, etc.) and analytical processing. We discuss here a prototype system known as System S from IBM Research which is currently being applied to a number of problem domains including environmental sensor-based solutions known as cyberphysical systems (Jain et al. 2006). These environments are now known as the object-based software integration frontiers for the physical world of measurements and actions (sensors/transducers and actuators of various sorts) and the traditional information technology environment. Cyberphysical systems are characterized by heterogeneous data types, telemetry, and computational hardware. In this section we provide an overview of the System S architecture and cite and several important enhancements.

System S has been designed with the goal of functioning in a distributed environment which experiences high loads and dynamic input. Distributed analytics are a key design point which support analytical functions ranging from filtering and traditional signal processing techniques to deep artificial intelligence (AI)-based analytics and complex modeling. Analytical algorithms are implemented through the construct know as processing elements (PEs). Incoming stream data objects (SDOs) are received by the input ports of the PE. Analytical algorithms within the PE process the stream and then may generate new SDOs for additional processing by one or more downstream PEs (see Fig. 1). A System S application may be thought of as a stream connected PE graph and the overall System S middleware platform provides services to run these applications.

As a middleware platform, System S has unique features such as dynamic application composition and stream discovery. Dynamic application composition allows the establish-

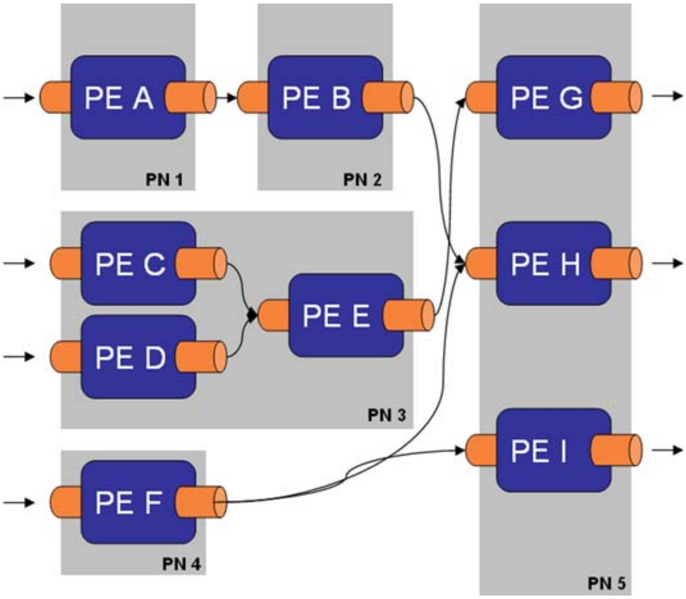

Fig. 1 Schematic representation of the System S distributed streaming analytical system. Interconnected processing elements (PE) reside on distributed processing nodes (PNs)

ment and termination of stream connections as new data sources and applications move in or out of the system realm. Reuse is supported for cross-platform/node collaboration as it is important for distributed, resource constrained environments. System S is a distributed stream processing platform that provides the architectural foundation to enable efficient analytical processing of large amounts of streaming data. Improvements to System S in terms of advanced distributed resource control have been pursued with ACES: Adaptive Control for Extreme-scale Stream processing systems which employed a two-tiered approach for adaptive, distributed resource control in environments with variable streaming data rates (Amini et al. 2006). ACES uses global optimization for the weighted throughput for processing graphs using input stream rates averaged over time. This is done dynamically as PEs are deployed or terminated, thereby ensuring responsiveness to changing workloads and resources. A resource controller then applies an adaptive, scalable and distributed optimization technique to optimize input and output rates of PEs factoring in the instantaneous processing rate of the PE with the goal of system stabilization for "bursty" workloads.

In Fig. 2 we describe the key architectural components of System S. The Dataflow Graph Manager (DGM) determines connections between PEs and matches the output port stream descriptions with input port specifications. Data transport in the distributed environment is handled by a set of daemons within the Data Fabric (DF). A daemon on each node of the system establishes transport connections between PEs and facilitates the movement of SDOs between "producer" PEs and "consumer" PEs. Runtime context and access to the middleware is performed by Processing Element Execution Containers (PECs). The PECs also provide a level of isolation required to prevent any possible corruption from user written PE code. PECs and DF daemons provide runtime statistics to the Resource Manager (RM) which makes global resource decisions for PEs as well as determines their placement.

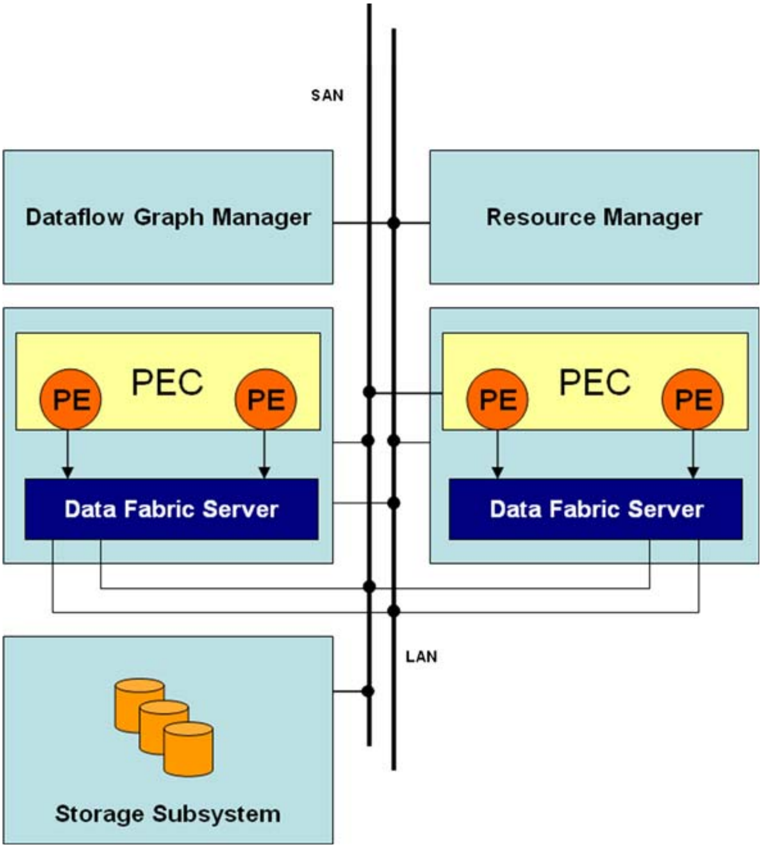

Fig. 2 The key components of System S that provide services to run stream applications (from (Jain et al. 2006))

Data flow graphs are used to represent applications and they are comprised of PEs that consume and produce data streams via respective input and output ports. These graphs are defined in a job configuration file and specify flow connections between the PEs. Stream data object data types acceptable to PEs (input and output) are specified in PE templates. PE connections are determined dynamically at runtime through the matching of SDO data types (defined in a single XML file) with flow specifications and this facilitates operations such as the discovery of new streams as they become available. This design adds flexibility and agility to the system as it prevents the need for the application developer to hardwire these connections. System S uses the Unstructured Information Management Architecture (UIMA) developed by IBM for unstructured information management solutions and made open source (IBM 2008).

To simplify the design and construction of streaming applications, a rapid application development environment for System S known as SPADE has recently been developed which utilizes an intermediate language for the composition of parallel and distributed flow graphs (Gedik et al. 2008). SPADE also provides a toolkit of stream processing operators for both scalar and vector processing and a robust library of stream adapters for stream ingestion and publication from/to outside sources.

4 Collaborative decision making

The real world is both unpredictable and unforgiving. Decisions often need to be made where the contributory evidence is uncertain, incomplete, contradictory and highly dynamic. The environmental arena is a case in point. This collage of factors is contributed to by the lossy nature of the environment and the fact that the sensed data (sensors) enable a set of beliefs to be derived about the current state of the environment. They are beliefs and not knowledge. Each sensed data stream offers insights along a specific geographic and temporal dimension and as such offers a partial inaccurate view. In reality such partial views must be aggregated or fused in order for example to achieve corroboration of a particular hypothesis. Such aggregation however is necessarily collaborative in nature demanding individual sensors cooperate in not only the diagnosis of a given situation, but rather a diagnosis that conforms to the given quality of service decision making constraints.

4.1 Multi-agent system approach

Collaborative Decision Making facilitates the establishment of partnerships to pool resources, jointly plan, and share expertise or knowledge so as that objectives may be achieved that go beyond the purview of any individual partner involved. This discussion focuses on the use of a Multi-Agent-System (MAS) to facilitate the collaborative decision making process. An agent is an autonomous system. That is, a system that is capable of deciding for itself what needs to be done in order to achieve its goals at runtime and without human intervention. Though endowed with particular responsibilities, each individual agent interacts with other agents, in a distributed manner, to fulfil the requirements of the application and in forming a MAS.

The Belief, Desire, Intention (BDI) model of agency (Rao and Georgeff 1995) is a mature and computationally tractable model of strong agenthood. Within the BDI model, beliefs represent possibly inaccurate information that an agent has of the world at a given instance in time; their underlying semantics conform to belief logics. Desires or goals represent a state of the world that an agent wishes to bring about. By providing agents with a means of identifying the purpose of a task, goals enable agents to recover from failures and to take advantage of unexpected events as they occur. An agent has a partial view of the world, and is resource bounded; it will therefore not capable of achieving all of its desires even if its desires are consistent. The agent must fix upon a subset of desires and commit resources to achieving them. This subset of desires represents the agent's intentions or commitments. Agents do not blindly commit to actions or plans, but revise their commitments at various stages throughout execution.

Traditionally, the BDI model was considered too computationally intensive for resource constrained mobile devices. In particular the emergence of a new generation of sensor devices which offer greater processing and memory capabilities as typified by the Stargate and Sun SPOT motes, the associated possibility of hosting restricted Java Virtual Machines as typified by the Squawk operating system and the development commercial Java based sensor software such as Sentilla all point to the possibility of hosting strong agents delivered via java on embedded sensing networks. This, combined with improvements in the efficiency of algorithms (Shen and O'Hare 2008), and the dissemination of good design practices mean that this argument no longer holds. It is for this reason that there are now a number of agent-based ambient systems. Of late there has been a rush toward the development of shrink wrapped agent platforms as exemplified by CourgarME (Wright and Moore 2003) and Agilla

(Fok et al. 2005). The agent-based component of Smart Bay has been developed using one such system, namely Agent Factory Micro Edition (AFME) (Muldoon et al. 2006). AFME distinguishes itself from other embedded agent frameworks, such as 3APL-M (Koch et al. 2005), LEAP (Berger et al. 2003), CougarME, and Sage-lite (Khalique et al. 2007), in a number of ways (see (Muldoon 2007)). It offers the smallest BDI software footprint available worldwide and represents the first BDI framework to be deployed on leaf nodes of a wireless sensor network, specifically Sun SPOT motes.

4.2 Agent factory micro edition

AFME is a minimised footprint deliberative agent platform that has been designed for use with resource constrained ubiquitous devices. It is based on Agent Factory, a pre-existing framework for the development of agents for desktop environments. AFME uses rules to define the conditions under which commitments will be desired and or adopted/retracted. These rules govern and encode agent behaviour. The following is an example of one such rule:

```
  a(?variable1), b(?variable2) > doSomething(?variable1,?
variable2);
```

If the agent adopts beliefs that match the a and b predicates, a commitment to doSomething will be adopted. The truth of the belief sentence is evaluated using resolution-based

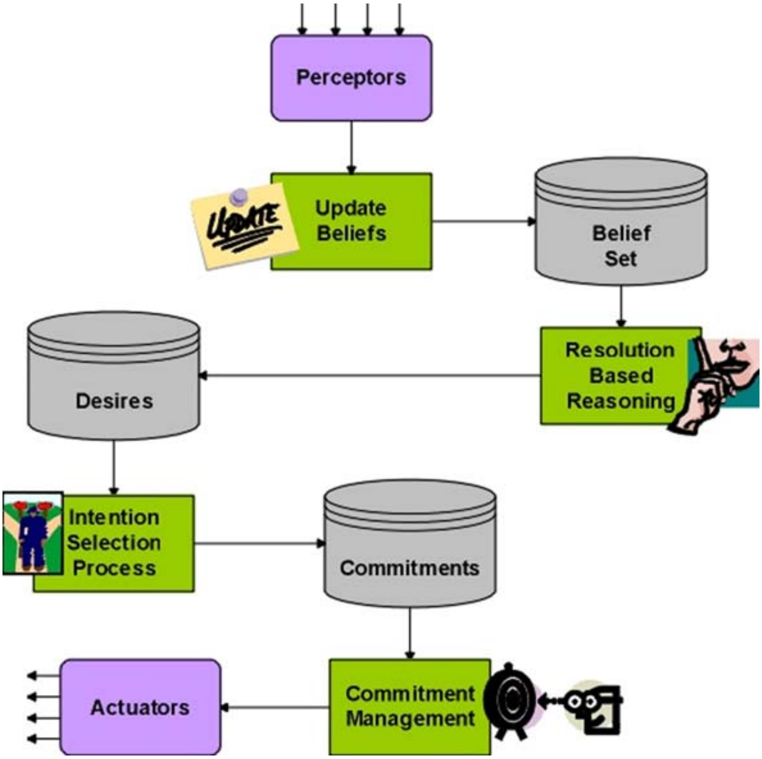

Fig. 3 AFME Control Process

reasoning. Fig. 3 illustrates the control process. In the control process, four functions are performed. First, perceptors that monitor the environment or agent's state are fired and the agent's belief set is updated. Second, the agent's desired states are identified. Third, a subset of commitments is chosen from the desire states in the intention selection process. Last, depending on the nature of the commitments adopted, various actuators are fired.

5 SmartBay: an environmental monitoring system

SmartBay, a pioneering project by the Marine Institute of Ireland, compromising a network of buoys linked by both fixed (cables) and wireless technologies. Each buoy incorporates a range of sensors and telemetry resulting in a real-time oceanographic monitoring system. One example of a sensor used in SmartBay for measuring phosphate levels is illustrated in Fig. 4. This sensor performs colorimetric chemistry and optical detection for measurement purposes. While monitoring standard oceanographic parameters, such as wave and tidal characteristics, it is hoped that SmartBay will provide data that will enable any perceived effects of climate change be measured in a meaningful way.

In collaboration with the SmartBay project, System S is being incorporated in an intelligent distributed cyber-infrastructure for the monitoring of the ocean and coastal areas of Ireland. SmartBay focuses on a number of knowledge domains with a particular emphasis on environmental variables and will support a diverse set of application areas and users (from research to commercial) using a number of data sources ranging from a myriad of sensors to external heterogeneous data sources (Fig. 5). Streaming data will support real-time

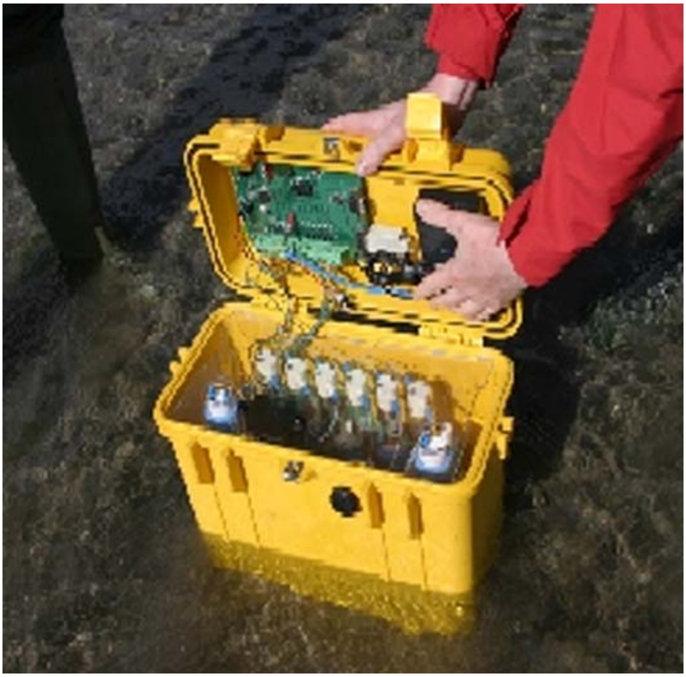

Fig. 4 One category of a sensor harnessed by SmartBay. This example measures phosphate levels

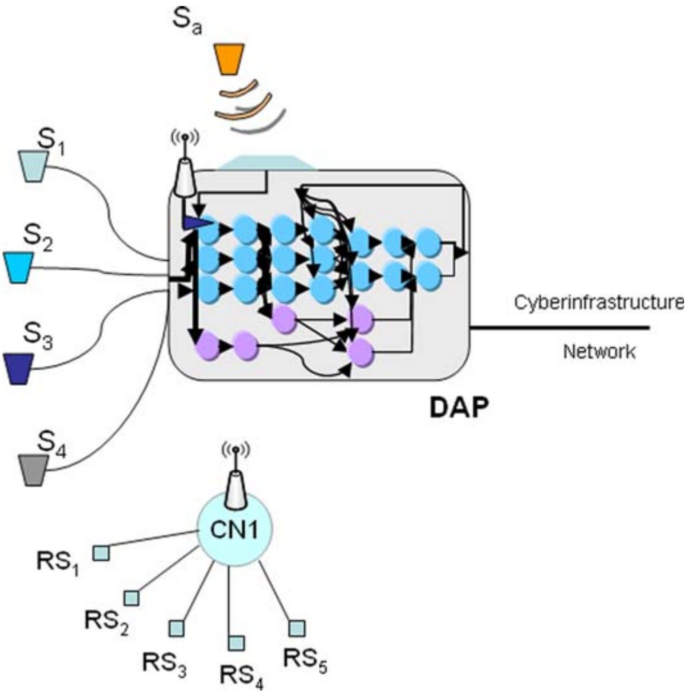

Fig. 5 A distributed analytical platform (DAP) will host System S streaming analytics for the SmartBay project. Multiple sensors will supply the inputs through direct connections (Sx) and various telemetry options including wireless (depicted as remote sensors at a concentrator node CN1), and acoustic (Sa) links such as those used in underwater remotely operated undersea vehicles (ROVs). The DAP is connected to the overall cyber-infrastructure via various communications options

monitoring and response applications. It is important to note that it is of great interest to share data between environmental observatories and to support some level of interoperability since these systems involve sizeable initial investments and require substantial resources for operations and maintenance. Given these factors, optimal utilization of these systems is an imperative as is the sharing of vast amounts of data/information to support scientific research efforts of local and global importance. A new approach for collaboration across stream processing systems has been reported as an enhancement to System S and is known as CLASP (Collaborating, Autonomous Stream Processing) (Branson et al. 2007). CLASP provides several benefits including improved sharing of raw data streams and derived data, thereby improving the breadth of analyses. In addition, CLASP supplies additional intelligence for resource sharing ("analytical load balancing") and reliability, the latter In terms of recover from job/application failures at distributed sites.

While System S can exist on the sensor devices themselves—it is vitally important to remember that other devices may not support such decision making entities. In reality a hybrid decision making architecture is required when dealing with a range of heterogeneous devices that have a broad spectrum of resources. SmartBay is no different and this gives rise to cooperation among these entities that reside on the devices, as discussed next.

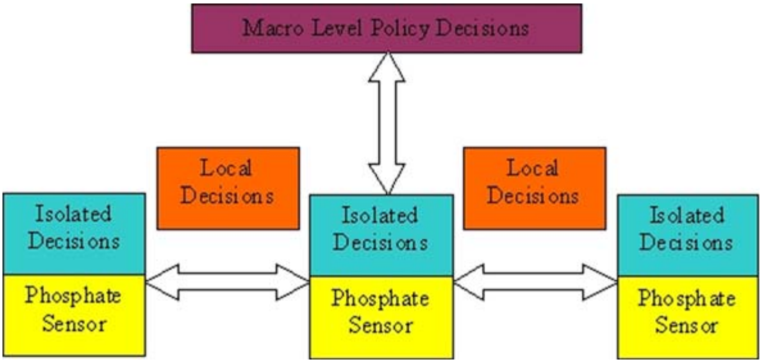

Fig. 6 Hybrid Decision-making within SmartBay

5.1 Hybrid decision-making

Within the SmartBay project, there are a number of areas where decisions can be taken to improve the performance of the system. These range from isolated or localised collaboration between nodes, to macro level decisions about the functioning of the system as a whole (Fig. 6).

Crucially the decision-making at each point is necessarily of a different type due to the distinct functions, quality of solution required and the resource limitations afforded by the hardware. This gives rise to a hybrid decision-making system in which multiple entities must coexist and cooperate to ensure the performance of the system is maintained. A number of possible metrics can be used to determine what exactly performance of the SmartBay system means:

- Longevity: Having a deployed network of sensors that will remain in operation for a few days will not suffice in most cases. Therefore decisions must be taken in order to conserve the limited power resources of the nodes.
- Sample Quality: Chemical sensors in general pose an idiosyncratic problem of device cleaning, and the phosphate sensor in use within SmartBay is no different. The quality of the next sample will depend on how well the device is flushed prior to taking the reading. This will inevitably lead to a reduced lifetime due to the power consumed in the cleaning cycle. Local decisions, informed by localized or indeed macro level policies can facilitate power conservation when sampling quality is not essential.

  ```
  !powerLevel(low,?level), flush, flushTime(?x) > flushSample
  (?x);
  ```

- Sampling Frequency: As with the previous issue, sampling frequency must be balanced with node lifetime in addition to Quality of Service concerns and this decision once again will be hybrid in nature. For instance, when interesting events are observed in the environment, the sampling rate of the node can increase.

  ```
  sampTime(?x), eventDetected(?y) > sampTime(?z);
  ```

- Transmission Frequency: In many cases it may not be essential to relay every sensed value back to some central repository, both bandwidth and power concerns can be factored into the decision here in order to decide on whether a nodes should transmit:

  ```
  transTime, processedData(?info), neightbour(?agent) >
  inform(?agent,?info);
  ```

- Analytics: Not only is there a decision about what analysis to apply to the data, but there can also be a decision about where that analysis is done. In some cases, it may be best suited to higher-level System S processing rather than local processing on a node. In addition to the QoS metrics, the agents will also have to receive policy decisions about its operation from the hybrid decision making components. Some examples of their behaviour is depicted below:

```
message(request,?sender,adoptPolicy(?policy)) >
processPolicy(?policy);

policy(?sampX, remoteAnalysis(?remoteAgent)) ,
sampleData (?sampX, ?data) , powerLevel(high,?level) >
inform (?remoteAgent,sampleData(?sampX,?data));

policy(?sampX, remoteAnalysis(?remoteAgent)) ,
sampleData (?sampX, ?data), powerLevel(low,?level) >
par(inform (?remoteAgent,sampleData(?sampX,?data)),
inform(?remoteAgent,policyChangeLocal),changeLocal);

policy(?sampX, localAnalysis),
sampleData(?sampX, ?data) >
processLocal(?data);
```

The agent contains a perceptor that is responsible for monitoring the power level. The local/remote analytic analysis is catered for in the rules above as they designate what agent will handle the analysis. The first rule above is responsible for processing policy requests from remote agents. The policy adopted by the agent will inform the sampling rate, transmission rate, and analytic processing decisions.

As can be seen from the previous list, decisions are very often taken on the basis of power resources. However, once the data is off the low level devices, other factors come into existence. Decisions about where best to locate the analytics engines can be made on the fly in order to accommodate various *hard* or *soft* real time requirements that the system may have. Additionally execution priorities may also be altered to ensure such deadlines are met.

5.2 Implementation

AFME has been designed for use with CLDC (Constrained Limited Device Configuration). This is the standard Java environment for mobile phones and the latest sensor devices. The Stargate and Viper board in use for the SmartBay project has capabilities far in excess of these devices and far exceed the requirements for executing AFME agents. They use CDC (Connected Device Configuration), which, although it is classified as J2ME, is in reality closer to standard Java than it is to CLDC (Muldoon 2007).

With the AFME compiler, Java classes are generated from a combination of an abstract agent platform description file, agent design files, and abstract template files that determine the imperative code generated. When agents were initially deployed on a Viper class device, a new template file was created with dependencies on CDC rather than CLDC. Additionally, new CDC platform services were created for message transport, discovery (yellow and white pages services), and migration. With these templates and services, AFME agents can be deployed on any device that supports CDC, such as the Viper or Stargate.

6 Conclusion

Environmental informatics is but one domain which typifies the need to process real-time heterogeneous data streams. The inherent complexities can benefit significantly from a collaboration reasoning approach in managing uncertainty, incompleteness and the presence of contradictory data. By their very nature such systems are often more complex in nature. To ensure a satisfactory Quality of Service (QoS), it is essential that appropriate decision making technologies be adopted, yielding a mix-and-match approach, influenced by the requirements of a given application. Recent developments in AI have enabled intelligent decision making to occur in resource-constrained devices such as WSNs. Such decision-making may be deliberative, combining fuzzy reasoning with classical BDI approaches, and may ultimately contribute to decision making elsewhere in the network or further up the application stack. With the proliferation of WSNs and their associated technologies, it can be expected that embedded decision making using a hybrid of techniques will become the norm as the complexity and scale of applications built on such technologies increase. This paper has described one such system Smartbay together with the particulars of the hybrid reasoning approach to deliver in situ decision making which combines stream based computing with multi-agent system techniques.

Acknowledgements The authors gratefully acknowledge the support of Science Foundation Ireland (SFI) under Grant No. 07/CE/11147 together with the support of the Marine Institute, Ireland.

References

Amini L, Jain N, Sehgal A, Silber J, Verscheure O (2006) Adaptive control of extreme-scale stream processing systems. In: ICDCS '06: Proceedings of the 26th IEEE international conference on distributed computing systems, IEEE Computer Society, Washington, DC, USA, p 71

Berger M, Rusitschka S, Toropov D, Watzke M, Schlichte M (2003) The development of the lightweight extensible agent platform. TILAB Journal—Special Issue on JADE 3(3):32–41

Branson M, Douglis F, Fawcett B, Liu Z, Riabov A, Ye F (2007) Clasp: olaborating, utonomous tream rocessing systems. In: Cerqueira R, Campbell RH (eds) Middleware, Springer, Lecture Notes in Computer Science, vol 4834:348–367

Chan C, Huang G (2003) Artificial intelligence for management and control of pollution minimization and mitigation processes. Eng Appl Artif Intell 16(16):75–90

Chen Q, Mynett AE (2006) Modelling algal blooms in the dutch coastal waters by integrated numerical and fuzzy cellular automata approaches. Ecol Model 199:73–81

Chen J, O'Grady MJ, O'Hare GMP (2006) Autonomy and intelligence—opportunistic service delivery in mobile computing, Lecture Notes in Artificial Intelligence (LNAI), vol 4253, Springer-Verlag, pp 1201–1207

Feng Y, Long T, Liu R, Guo J, Liu X (2003) Study on biological filter anaerobic wastewater treatment hybrid intelligent control. Proceedings of the 4th international conference on control and automation (ICCA'03) pp 441–445

Fleming G, van der Merwe M, McFerren G (2007) Fuzzy expert systems and gis for cholera health risk prediction in southern africa. Environ Model Softw 22(4):442–448

Fok CL, Roman GC, Lu C (2005) Mobile agent middleware for sensor networks: an application case study. In: Proc. of the 4th Int. Conf. on Information Processing in Sensor Networks (IPSN'05), IEEE, pp 382–387

Fregene K, Kennedy D, Madhavan R, Parker LE, Wang D (2005) A class of intelligent agents for coordinated control of outdoor terrain mapping UGVs. Eng Appl Artif Intell 5

Gedik B, Andrade H, Wu KL, Yu PS, Doo M (2008) Spade: the system s declarative stream processing engine. In: SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on management of data, ACM, New York, NY, USA, pp 1123–1134

Heaton JT (2005) Introduction to neural networks with Java. Heaton Research Inc.

IBM (2008) Unstructured information management architecture (UIMA). http://www.research.ibm.com/UIMA/

Jain N, Amini L, Andrade H, King R, Park Y, Selo P, Venkatramani C (2006) Design, implementation, and evaluation of the linear road bnchmark on the stream processing core. In: SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on management of data, ACM, New York, NY, USA, pp 431–442

Keegan S, O'Hare GMP, O'Grady MJ (2008) Easishop: ambient intelligence assists everyday shopping. Inf Sci 178(3):588–611

Khalique S, Farooq S, Ahmad HF, Suguri H, Ali A (2007) Sage-lite: an architecture and implementation of light weight multiagent system. ISADS 0:239–244, http://doi.ieeecomputersociety.org/10.1109/ISADS.2007.68

Koch F, Meyer JJ, Dignum F, Rahwan I (2005) Programming deliberative agents for mobile services: the 3APL-M platform. AAMAS'05 Workshop on Programming Multi-Agent Systems (ProMAS05) pp 222–235

Muldoon C (2007) An agent framework for ubiquitous services. PhD thesis, School of Computer Science and Informatics, Dublin, Ireland

Muldoon C, O'Hare GMP, Phelan D, Strahan R, Collier RW (2003) ACCESS: an Agent architecture for ubiquitous service delivery. In: Proceedings 7th international workshop on Cooperative Information Agents (CIA), Springer, Helsinki, Lecture Notes in Computer Science, vol 2782, pp 1–15

Muldoon C, O'Hare GMP, Collier RW, O'Grady MJ (2006) Agent factory micro edition: a framework for ambient applications. In: Intelligent agents in computing systems, Springer, Reading, UK, Lecture Notes in Computer Science, vol 3993, pp 727–734

O'Grady MJ, O'Hare GMP (2004) Gulliver's genie: agency mobility adaptivity. Comput Graph 28(5):677–689

O'Grady MJ, O'Hare GMP (2005) Mobile devices and intelligent agents: towards a new generation of applications and services. Inf Sci 171(4):335–353

O'Hare GMP, O'Grady MJ, Muldoon C, Bradley JF (2006) Embedded agents: a paradigm for mobile services. Int J Web Grid Serv 2(4):379–405

Rao AS, Georgeff MP (1995) BDI Agents: from theory to practice. Proceedings of the first international conference on multi-agent systems (ICMAS'95) pp 312–319

Recknagel F, Petzoldt T, Jaeke O, Krusche F (1994) Hybrid expert system delaqua—a toolkit for water quality control of lakes and reservoirs. Ecol Model 71(1-3):17–36

Shen S, O'Hare GMP (2008) Fuzzy decision making through energy-aware and utility agents within wireless sensor networks. Artificial Intelligence Review

Taha I, Ghosh J (1995) A hybrid intelligent architecture and its application to water reservoir control. Int J Smart Eng Syst 1:59–75

Wright W, Moore D (2003) Design considerations for multiagent systems on very small platforms. In: AAMAS '03: Proceedings of the second international joint conference on autonomous agents and multiagent systems, ACM Press, New York, NY, USA, pp 1160–1161

Wu F (1998) Simulating urban encroachment on rural land with fuzzy-logic-controlled cellular automata in a geographical information system. J Environ Manag 53(4):293–308