Presentation 10
Agnostic Questions
Paper: “Autonomic Computing: An Overview”, by M. Parashar and S. Hariri
Presenter: Alejandro Simon
Agnostic: Joseph Cilli

Question 1
Being that the ACS paradigm is modeled after the human nervous system, is it accurate to conclude that the paradigm is fallible when you consider that a system might not be able to self-heal an unknown/unforeseen issue just as the human biological system may not be able to self-heal because it is unaware of specific infections and its inability to learn unless combined with artificial intelligence?

Answer 1
I think it is accurate to say that the self-healing capabilities of an ACS depend on the sophistication of the algorithm. Even with the use of artificial intelligence, the possibility of the system being exposed to an unforeseen issue will always exist. Therefore, consulting an expert is a wise course of action.

Discussion
Dr. Sadjadi used an example of the amount of vaccination shots given today versus when he was a child. The point of the example was to show society is familiar with new viruses or problems. The shots are used as self-protection to have your body defend itself. The hope is the dynamically change software as it is executing, so the existence would not have to shutdown and can learn as it goes.

Question 2
The authors’ state an ACS requires sensor channels to sense changes and motor channels to react; however, there is never any mention of the overhead involved with sensing, analysis or reaction. Is there a significant cost to these requirements that is overlooked in the paper?

Answer 2
There is always computational overhead involved in executing extra code. However, use of threads and performance analysis of the application to ensure it meets the contractual Quality of Service should ensure that the overhead introduced by the monitoring and healing code does not appear to hinder the application. There is a substantial cost in terms of programming time when implementing these requirements.

Discussion
If you have a real-time system, by adding monitors and/or actuating components the system must still function in real-time otherwise the system is a failure. The example given is using TRAPJ and making a system adapt ready. This is done to monitor some aspects of the system that are of interest. In the absence of need for adaptation, your system should work as the original system. If adaptation is needed, the value of that adaptation should compensate for the overhead.

Question 3
The idea of ACS seems like a viable solution to handle complex systems. Yet there have been instances where an ACS has caused more work for an administrator. Have there been any studies to compare system performance using solely human interaction with system performance using an ACS?

Answer 3
Not familiar with a specific study comparing the performance of human interaction vs. an ACS.

Discussion
If your system is not designed with care, your complex system is now incorporating its self-management behavior. That could mean an even more complex system. Research exists (transparent shaping) to address separating the development of self-management from the business logic of applications. If not careful, the autonomic version of your system could be more complex than the original system.

Question 4
Once high level policies are defined by a human, how would an ACS handle conflict resolution between contradictory self management aspects?

Answer 4
Several algorithms can be implemented to implement conflict resolution. The most trivial one is to require human input. Another option is to implement either one of the conflicting strategies, monitor and record its effect in the system. After several iterations, choose the one with the highest success rate. This approach would mimic the way humans learn by making choices with incomplete information and learning from their mistakes.

Discussion
An example given by the agnostic was a system the opens a port for self-optimization versus self-protection trying to close that port. There is no concrete way to solve this except to have a high level policy to handle priority and conflict resolution. This is still an open research problem.

Question 5
The authors' state that one of the research challenges is autonomic application & system architecture. Since the proposed architectures deal with autonomic element communication, can one infer that such architectures would be OS and/or machine specific?

Answer 5
Not necessarily, with the advert of managed languages and open communication standards such as Java, .NET languages, HTTP and SOAP, such architectures need not be OS nor machine specific.

Discussion
An ACS can be self-managed within itself; that is, the platform does not matter if the systems self- aspects are performed within the context of the application.

Question 6
Once a human sets high level policies, how can a non-programming administrator determine if the self-management aspects of a system are performing properly/efficiently?

Answer 6
A non-programming administrator is limited to monitoring the system as a black box. Such monitoring can include query execution times, and other statistic reported by the self-management modules.

Discussion
Dr. Sadjadi stated the whole process starts with monitoring and analyzing the behavior of the system to know if it is running within equilibrium. The question was redefined to read as 'How does the administrator know if the policies he/she is defining are the best policies to ensure system efficiency'. This was answered as the administrator should have intimate knowledge of the system and should be able to complete the tasks being allocated to the system. The tasks are not being allocated to the system because the administrator can't complete them; on the contrary, they are pushed down to the system as a time saving mechanism.

Question 7
Clearly ACS has a broad use in many industries; however, the authors' examples of existing projects mostly focus on data management systems. Was the current concept of ACS spawned by DBMS or does it simply lend itself to that arena?

Answer 7
Each author would choose examples that are most familiar to them. The concept of ACS applies to DBMS given the complexity of such systems and the familiarity of Computer Science majors with such systems. ACS however, are not limited to DBMS as the examples below show:

-"An Al tool for supervising substations", Melvin Ayala, S. Galdenoro Botura, J. Oscar Maldonado, IEEE POTENTIALS, 2002, VOL 20; PART 5, pages 13-18
-"http://www.ForexLab.NET", A mechanical trading software.

Discussion
The autonomic system defined by the IBM initiative was developed from IBM products like DB2 and services; however, the other elements of self-management have other sources.

Question 8
The authors' conclude that achieving overall autonomic behaviors remain an open and significant challenge, which can be accomplished, in part, by open industry standards; however, a lone standards organization body has yet to be defined. How can the stated challenges be met when multiple governing bodies exist?

Answer 8
The same way problems have been solved in the past, several groups implement a solution to the problem, a particular implementation becomes a de-facto standard and the market determines which solution fits each particular niche.

Discussion
Currently, there are many standards organizations developing different standards for many varying reasons. This is viewed as a good thing as it promotes multiple points of view. Ideally, a marriage of standards would be beneficial.

Additional Discussions
An issue was raised pertaining to the two control loops in relation to which loop was more important as shown in the presented diagram. The discussion concerned the interpretation of the internal loop and external loop. Basically, internal loops deal with finer-grain changes; whereas, the external loop would concentrate on more coarse-grain changes. Finer-grain changes, which are environment changes, may allow effectors change the environment back or fight against it. Coarse-grain changes might be too big of a change and would be out of the scope of control to bring back to original or stable state; therefore the system should be able to adapt itself.

Dr. Sadjadi posed a question to the class: How is it possible to adapt to an unanticipated change in the future? Many answered by stating human interaction based on messages sent from the system could be used. The use of patches to a system could be a way to adapt to an unanticipated change. A system can be designed so that every change can be made using dynamic adaptation; however, if your system intercepted and redirected every interaction the system performance would suffer to the point of unusable. In conclusion, static code is useful for the things you are sure about…we are not making everything autonomic.

Tariq King questioned why the term open was used to define an ACS and why his application had to be portable. Both authors of the paper are from GRID computing field, so they would love for self-healing by migrating one component from one system to another; however, openness is not as important as the other aspects defined by the presenter. By definition, if you implement one of the stated aspects (self-configuring, self-healing, self-protecting, etc.) then your system is autonomic. Therefore if a system is open, that does not imply autonomic. The autonomic system must have a feedback loop.

In addition to the stated challenges of autonomic computing, human computer interaction was identified as a challenge. Validation was also mentioned as a challenge. Prematurely releasing of software for financial gains before accurate testing is conducted was viewed as a societal challenge.

The agnostic asked if anyone was aware of any research in which scientists were using things learned by ACS to solve human nervous system problems (ie. paralysis). No one was aware of such research; however, discussions ensued pertaining to microchips implanted in the brain of paralyzed individuals to interpret brain signals which could move a mouse cursor. Dr. Sadjadi requested more information about this topic be forwarded to him.