A Technique for Measuring the Level of Autonomicity (LoA) of Self-managing Systems

Thaddeus Eze

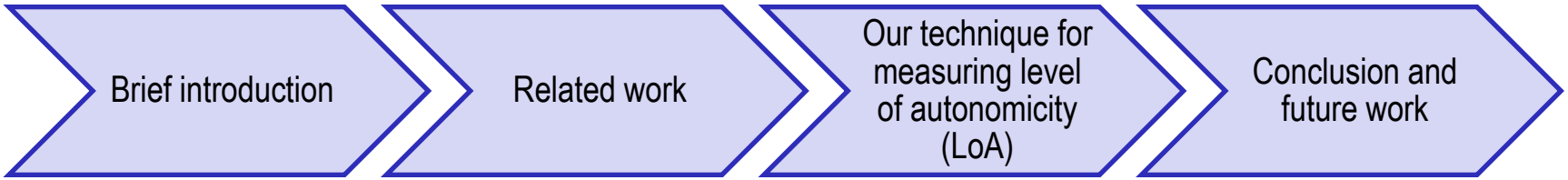
Richard Anthony

Chris Walshaw

Alan Soper

… for ICAS 2012, March 25-30, 2012 - St. Maarten, Netherlands Antilles

Outline



- autonomic computing
- what is and
- why LoA?

- current techniques for measuring LoA
- challenges

- our proposal
- strength and challenges

- where we have been,
- where we are and
- where we are going

Brief Introduction: Autonomic Computing

- Autonomic Computing (AC) seeks to build self-managing systems (aka Autonomic Systems) with four basic self-* functionalities to address the growing complexity of managing computing systems.
  - Autonomic Systems (AS) are defined, at least in the AC white paper [1], by four autonomic functionalities (the self-*) and as there are yet no consensus on the emerging functionalities, we stick with this initial and widely accepted definition.

Self-*:

- self-Configuring
- self-Healing
- self-Optimising
- self-Protecting

} Generally accepted core functionalities of an AS

Brief Introduction: What is and why LoA?

Autonomicity: the ability of a system to pursue its goal with minimal (or no) external interference in the form of configuration or control. Then, the extent of this ***interference*** defines LoA.

$$\Rightarrow \quad interference \propto \frac{1}{LoA}$$

Autonomic technology is advancing at a high rate, yet there are no universal standards for the technology itself and the design methods used:

> A vital step in the path towards certifiable AS is to introduce robust techniques by which the systems can be described in universal language, starting with a description of, and means to measure the extent of autonomicity exhibited by a particular system [2].

Related work

• Scale-based approach: uses a scale of **(1– n)** to define a system's LoA where **'1'** is the lowest autonomic level usually describing a state of least machine involvement in decision-making and **'n'** the highest autonomic level describing a state of least human involvement. [3][4][5]

- approach is qualitative and does not discriminate between behaviour types.

• IBM's 5 levels of automation: classifies autonomic systems according to 5 levels.
IBM Autonomic Computing White Paper (2005)

- too narrowly defined and the differentiation between levels is too vague to describe the diversity of self-management.

• Agent autonomy attributes based: identifies autonomy attributes and then defines a set of measures for each attribute. [6]

- attributes used do not define core autonomic functionalities

We posit that a more appropriate approach should comprise both qualitative and quantitative measures and should consider the core autonomic functionalities

LoA: Proposed technique –Defining an AS

Given that any AS is defined by the four self-* autonomic functionalities (represented as C, H, O and P), the following expression gives the possible combinations of the functionalities:

$$\sum_{r=1}^{4} {}^{n}C_{r} \qquad \rightarrow \text{ 16 combinations}$$

Where (***n* = 4**) is the number of functionalities (the CHOP) and ***r*** is a category of the possibilities (a specific implementation combination of the functionalities)

$\Longrightarrow$ Any system deemed autonomic can be defined in one of 15 ways (excluding zero value which is a special case and considered non autonomic –line 16 in fig. 1)

LoA: Proposed technique –Defining an AS

An AS is defined based on its achievement of the CHOP functionalities [7]

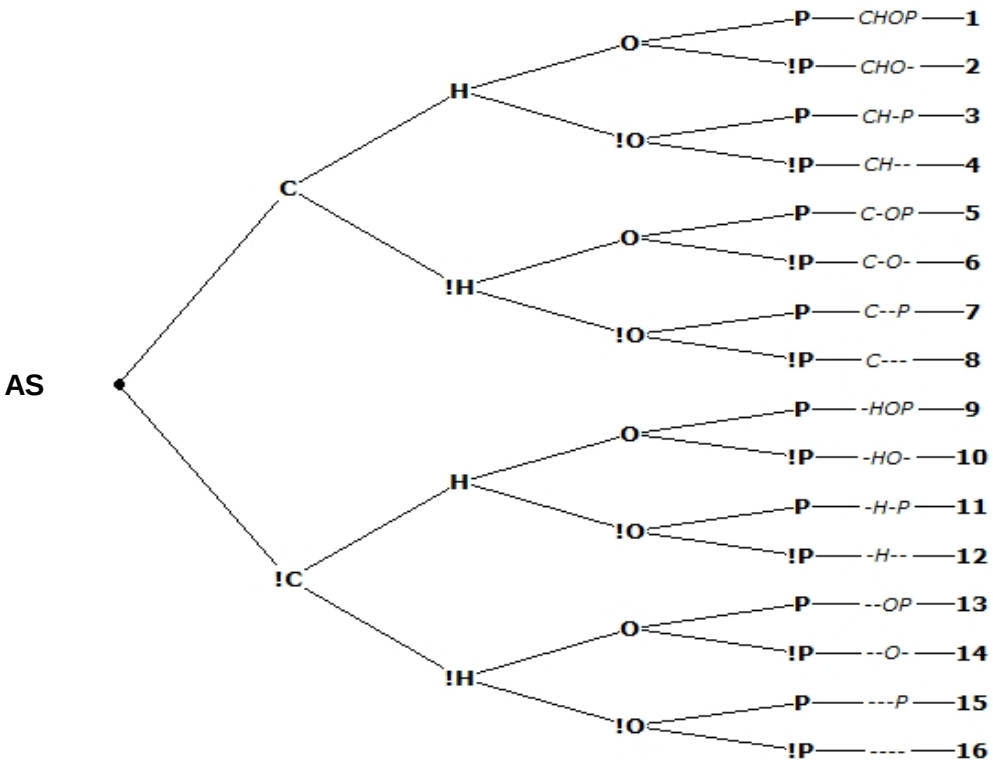


Figure 1: Combination of autonomic functionalities.

Key:
CHOP = full autonomic (in terms of self-*)
C–O– = system demonstrates only *Configuring* & *Optimising* functionalities
–HO– = only *Healing* and *Optimising* functionalities

LoA: Proposed technique –Measuring LoA

It then follows that:

An AS is defined based on its achievement of the CHOP functionalities.

Each functionality is defined by a set of metrics (which can be application dependent)

An autonomic value contribution is assigned to each functionality which is spread across the set of metrics for that functionality.

$\Longrightarrow$ let the maximum LoA value for an AS be ***M***,

then a system will have a LoA value of ***N (0 ≤ N ≤ M)*** and

each functionality contributing a value in the range ***(0 ≤ x ≤ M/4)***, while

each metric of each functionality contributes ***(M/4)/m (m ≠ 0)*** autonomic value

*where **m** is the number of identified metrics defining a particular functionality, and the constant **4** represents four CHOP functionalities.*

LoA: Proposed technique –Measuring LoA

So if we define autonomic metrics for each of the functionalities, then **the sum of the autonomicity in each of the constituent functionalities**, for a particular AS, gives the system's LoA:

$$LoA = \sum_{i=1}^{m_c}[c_i] + \sum_{j=1}^{m_h}[h_j] + \sum_{k=1}^{m_o}[o_k] + \sum_{l=1}^{m_p}[p_l]$$

Where subscripted ***m*** is the number of identified metrics for the respective functionalities. $\boldsymbol{c_i}$, $\boldsymbol{h_j}$, $\boldsymbol{o_k}$ and $\boldsymbol{p_l}$ are the autonomic metric contributions of the functionalities. These can be composed of functions of different measures but are normalised to yield autonomic values.

We can apply LoA to any of the scale-based approaches to qualitatively interpret our results.

e.g, if we choose the **8-level** autonomy assessment scale in [3], then we will assign ***M = 8*** and interpret the LoA result within the corresponding scale:

> a value in the range **(2 < LoA ≤ 3)** will be interpreted within level **3** on the scale which is '*human shadows computer*'

LoA: Proposed technique –Metrics examples

- Though metrics are application domain dependent, we present some generic and specific metrics. These serve as examples for each of the functionalities.

***Self-Configuring*:** When a new component is introduced into an AS it registers itself so that other components can easily interact with it. Interoperability ratio (*I*) is a measure of self-configuration, measured as ratio of the actual number of components ($_{actual}n_i$) successfully interacting with the new component (after configuration) to the number of components expected ($_{expected}n_i$) to interact with the new component.

$$I = \sum_{1}^{i} \frac{{}_{actual}n_i}{{}_{expected}n_i}$$

Self-Optimising: If we consider a resource allocation environment where service level space and demand space are expressed using utility functions as **U(s)** and **U(d)** respectively, we can define resource allocation ratio (R) as a self-optimising metric:

$$R = \frac{U(s)}{U(d)}$$

The ultimate goal of the system is always top priority

LoA: Proposed technique –Metrics examples

***Self-Healing*:** We define *reaction time T* as a metric for self-healing capability. This is crucial to the reliability of a system. If a change occurs at time t_a and the system is able to detect and work out a new configuration and ready to adapt at time t_b then *T* defines the reaction time. (Average is taken instead where variations of *T* are possible).

$$T = t_b - t_a$$

***Self- Protecting*:** We define *ability to detect repeat events (E)* as a self-protecting metric. *E* is a Boolean value (True indicates the manager is able to stop a repeating problem and False otherwise). If we choose two samples of $\{p_{ij}\}$ (a log of all identified trends and corresponding problems –if *i^{th} trend leads to j^{th} problem*) at different times (t_1 and t_2) then we can define *E.* (Different trends may lead to the same problem but a repeated trend-problem combination indicates a failure of the system to prevent a reoccurrence).

$$E = True \quad \forall_{ij} \quad if \quad \{p_{ij}\}_{t1} \cap \{p_{ij}\}_{t2} = \varnothing$$

LoA: Proposed technique –Case study

Our case study is *Dynamic Qualitative Sensor Selection System* (DQSSS), based on work in [8]. The goal of DQSSS is to *dynamically select a sensor (amongst many) based on continuously variable qualitative characteristics* (e.g., signal quality and noise levels).

By definition self-configuration, optimization and healing are of importance to this system (*r=3*).

LoA: Proposed technique –Case study

The DQSSS is presented in three progressive stages (addressing one challenge at a time as the sensor selection problem is explored in increasing level of detail) represented as systems A, B and C:

- ✓ All three systems are able to differentiate sensors by their signal characteristics using *utility functions*.
- ✓ Systems B and C are able to generate trends in signal quality using *trend analysis* logic.
- ✓ Only system C ensures stability (avoiding unhealthy oscillation in sensor selection) by implementing *dead zone* logic,
- ✓ while none of the systems has a way of detecting a failed sensor.

LoA: Proposed technique –Case study

TABLE I: REPRESENTATION OF THE DQSSS

Characteristics (metrics)	**Contributing CHOP**	**Sys A**	**Sys B**	**Sys C**
Continuous	C	√	√	√
Unsupervised	C	√	√	√
Trends examination	O	-	√	√
Stability	O	-	-	√
Dynamic (logic switching)	O	-	-	√
Signal characteristics	C	√	√	√
Signal differentiation	C	√	√	√
Failure sensitivity (sensors)	H	-	-	-
Robust (fault tolerance)	H	-	-	√

We adopt Ryan *et al* 8-level autonomy assessment scale in [3] as a way of qualitatively interpreting our results. In keeping with this we adopt the arbitrary value **8** as the maximum LoA implying that each CHOP contributes an autonomic value in the range **($0 \leq x \leq 2$)** spread across its metrics.

Normalizing the identified metrics in Table I (the numbers of metrics in each category are: **C=4, H=2, O=3**) in the autonomic value range **($0 \leq x \leq 2$)** gives the result in Table II.

LoA: Proposed technique –Case study

TABLE II: ANALYSIS RESULT

	Sys A	Sys B	Sys C
C	2	2	2
H	0	0	1
O	0	0.67	2
LoA	**2**	**2.67**	**5**

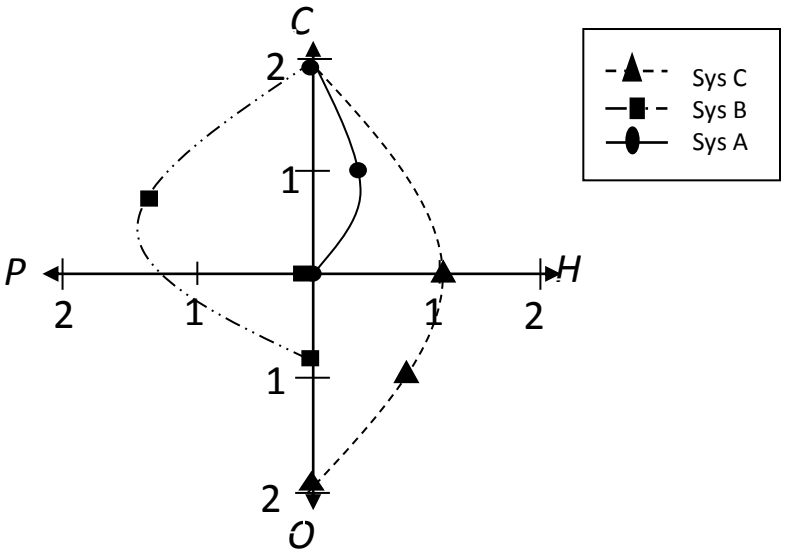


Figure 2: LoA representation of systems A, B & C in the four CHOP domains.

Recall that only three functionalities **(CHO–)** are of importance here which means maximum LoA value of **6**.

Out of this maximum value, systems A, B and C achieved the values 2 (i.e., 33%), 2.67 (i.e., 45%) and 5 (i.e., 83%) respectively.

LoA: Proposed technique –Case study

Tying this result with the 8-level autonomy assessment scale:

System A falls within level 2 of the scale which points to a situation where '*computer shadows human*' in the self-management process.

This indicates that system A only has a narrow envelope of environmental conditions in which it is both autonomic and returns satisfactory behaviour.

System B tends toward level 3 on the scale which is '*human shadows computer*'

which translates into a wider operational envelope, but once the limits of that envelope are reached human input is needed in the form of retuning, or manual override in the case of oscillation, which for example system C can deal with autonomicaly.

LoA: Proposed technique –Case study

System C falls within level 5, which points to '*collaboration with reduced human intervention*'.

This indicates that C is sufficiently sophisticated to operate autonomicaly and yield satisfactory results under almost all perceivable operating circumstances.

LoA: Proposed technique

Strength

- Quantitative & qualitative
  - Quantitative results can be qualitatively explained using any of the scale-based approaches
- Self-* orientated
  - Based on the core autonomic functionalities
- Generic
  - Applies to any AS
- Based on ISO/IEC 9126-1 standard*

Challenge

- Assumes orthogonality
  - i.e. Assumes functionalities do not overlap
- Metrics can be difficult to define and normalise
  - Requires standard definitions
- System yet to be fully built

* ISO/IEC 9126-1 standard decomposes overall software product quality into characteristics, sub characteristics (attributes) and associated measures [9].

Conclusion

The benefit of analyzing Autonomic Systems in terms of their extent of autonomicity not only offers a path to Autonomic Systems' certification as stated earlier, it also offers a way of comparing alternate systems, and also facilitates a proper description of these systems to users.

This is the first vital step in the groundwork for the introduction of appropriate standards for AC, in terms of technologies and the composition of functionality as well as validation methodologies.

We must **now** begin to design autonomic systems with **trustworthiness** in mind

References

[1] IBM Autonomic Computing White Paper, *An architectural blueprint for autonomic computing*. 3rd edition, June 2005

[2] Thaddeus O. Eze, Richard J. Anthony, Chris Walshaw and Alan Soper. *The Challenge of Validation for Autonomic and Self-Managing Systems.* In proceedings of The 7th International Conference on Autonomic and Autonomous Systems (ICAS), May 22-27, 2011 – Venice/Mestre, Italy

[3] Ryan W. Proud, Jeremy J. Hart, and Richard B. Mrozinski. Methods for Determining the Level of Autonomy to Design into a Human Spaceflight Vehicle: A Function Specific Approach. http://handle.dtic.mil/100.2/ADA515467 accessed 13/03/12

[4] Clough B T. Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway? In Proceedings of PerMis Workshop, pp 1–7. NIST, Gaithersburg, MD, 2002.

[5] Sheridan T. B.. *Telerobotics, Automation, and Human Supervisory Control. The MIT Press. Cambridge, MA, USA 1992. ISBN:0-262-19316-7*

[6] Fernando Alonso, José Fuertes, Löic Martínez, and Héctor Soza. *Towards a Set of Measures for Evaluating Software Agent Autonomy. In proceedings of 8th Mexican Int'l Conference on Artificial Intelligence (MICAI), 2009*

[7] Bantz D. F. Bisdikian, C. Challener, D. Karidis, J. P. Mastrianni, S. Mohindra, A. Shea, D. G. and Vanover, M.. *Autonomic Personal Computing*. IBM Systems Journal, Vol 42, No 1, 2003

[8] R.J. Anthony. *Policy-based autonomic computing with integral support for self-stabilisation*, Int. Journal of Autonomic Computing, Vol. 1, No. 1, pp.1–33. 2009

[9] ISO/IEC 9126-1:2001(E), Software engineering — Product quality — Part 1: Quality model

thank you

questions?