

# **Viable System Agent**

**Robert Bruce Carleton**

<b>Introduction</b>	<b>3</b>
<b>History</b>	<b>3</b>
<b>Project status</b>	<b>3</b>
<b>Subsystems</b>	<b>4</b>
<b>Measuring achievement</b>	<b>6</b>
<b>Alerts and audits</b>	<b>6</b>
<b>References</b>	<b>7</b>

## **Introduction**

The Viable System Agent, or VSA is a software implementation of the Viable System Model (VSM), a model for autonomous systems developed by Stafford Beer. The VSA provides structure, communications, auditing and alerting as defined in the VSM. The VSA is intended to facilitate creating and managing autonomous software systems. Think of the VSA as common instrumentation and communications for autonomous agents.

## **History**

One of the best known projects making use of the VSM was an attempted real-time computer-controlled planned economy in the years 1970 to 1973. It was called Project Cybersyn. The system used a telex network, applications running on an IBM 360 computer and a specially designed operations room for decision making. Given the advances in computers and networks, the VSM has the potential to be far more automated today.

## **Project status**

Most of the work on the VSA has been on the autonomies in systems 1, 2 and 3. They are presently available only as API's. There is no user interface, beyond the Squeak and Pharo development environments. See the class documentation for more information. The existing API interfaces are stable for the foreseeable future. Experience gained from developing specific applications will be the driver for future interface changes.

# Subsystems

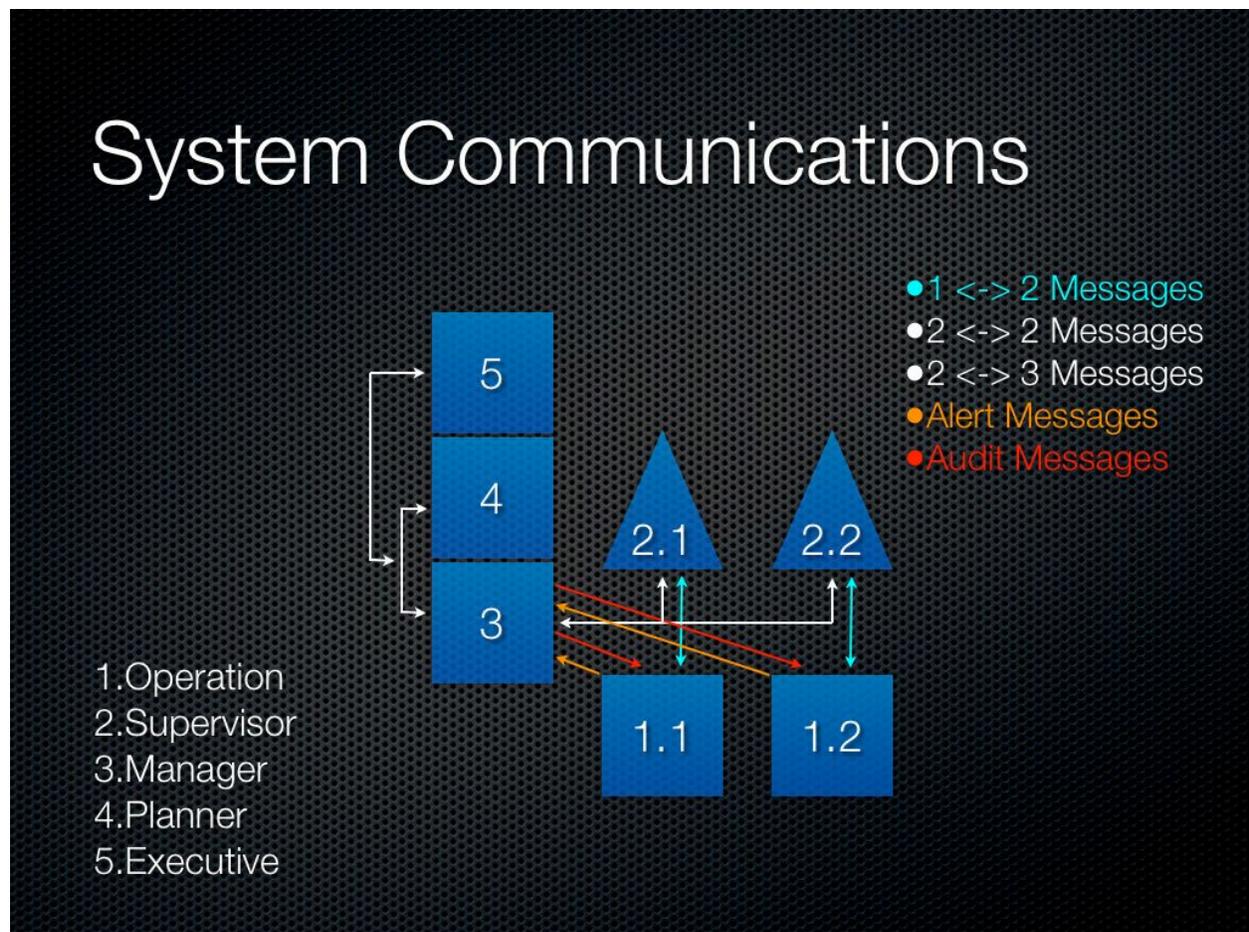
A VSA includes five subsystems.

## Corporate

System 5 (the executive) maintains system identity and controls communications throughout the VSA. System 4 (the planner) monitors the environment and plans for the future. System 3 (the manager) controls and reports on subordinate operation activities. Together, systems 3, 4 and 5 act as a triumvirate guiding the entire system.

## Autonomic

A VSA may have one or more subordinate operations, composed of supervisor/operation pairs. A supervisor (system 2) provides coordination and communications functions. An operation (system 1) is a primary activity of the VSA. These pairs are managed by system 3.



## **Recursion**

The entire VSA is also an operation and acts as a wrapper around all the other sub-systems. It represents the upward facing activity of the VSA as a whole, and may be subordinate to another VSA.

## **Notes**

By default, VSA instances start their own manager (3), planner (4), and executive (5) instances. This is to support Beer's ideas of cybernetic isomorphism. The idea is that all operations function as fully viable systems in their own right, possibly having embedded operations.

The system 2 runs as a peer instance with the 3, 4 and 5 instances inside a VSA. The system 2 is paired with a VSA is part of the upward facing communications with the next tier of control.

# Measuring achievement

A big part of the VSM is measuring the achievement of operations. Beer breaks this out into three measures reflecting the "here and now" compared to the "there and then". The measures can represent "more is better" numbers such as net income or web impressions served. They can also represent "less is better" numbers such as man hours used or amps consumed.

## Achievement measures

- Actuality - What we are doing now, with existing resources, under existing constraints
- Capability - What we could be doing right now with existing resources, under existing constraints
- Potentiality - What we could be doing by developing our resources and removing constraints

## Achievement calculations

Beer provides calculations to aid in monitoring overall performance of an operation. In addition to the familiar productivity calculation, other calculations allow actuality and capability to be compared in respect to potential growth. Profit center calculations are shown. The terms may be reversed for a cost center.

- Productivity - The ratio of actuality and capability
- Latency - The ratio of capability and potentiality
- Performance - The ratio of actuality and potentiality, and also the product of latency and productivity

The latency calculation can determine if productivity is being increased at the expense of capability. This might be particularly useful in times of growth.

## Alerts and audits

Alerts are sent by an operation to the manager when actuality deviates by a statistically significant amount from capability (productivity). Audit messages will be sent by the manager to operations to audit productivity (actuality/capability).

# References

## Primary

- Beer, Stafford. The Heart of Enterprise (1979). Chichester: John Wiley & Sons.
- Beer, Stafford. Brain of the Firm (2nd ed./1981). Chichester: John Wiley & Sons.

## Additional

- Bayne, Jay S. Cyberspatial Mechanics, IEEE Transactions on Systems, Man, and Cybernetics-part b: Cybernetics, Vol. 38, No. 3, June 2008. IEEE.
- Block, F., Cottle, S., Murray, R. Graham and Dodd's Security Analysis (5th ed./1988). New York: McGraw-Hill Inc.
- Harrison, P. J., and Stevens, C. F., 'A Bayesian Approach to Short-term Forecasting', Operational Research Quarterly, 22, No. 4. December 1971.
- Harrison, P. J., and Stevens, C. F., 'Bayesian Forecasting', Journal of the Royal Statistical Society, Series B (Methodological), 38, No.3, 1976.
- Herring, Charles E. The Viable System Model for Software (2000). Brisbane, Australia: University of Queensland.
- Herring, Charles E. The Pattern of the Viable System and its Language (2001). Brisbane, Australia: University of Queensland.
- Herring, Charles E. Groove: A Case Study in Adaptive Architecture (2001). Brisbane, Australia: University of Queensland.
- McMillan, Lawrence G. Options as a Strategic Investment (4th ed./2002). New York: New York Institute of Finance.
- Medina, Eden. Cybernetic Revolutionaries: Technology and Politics in Allende's Chile (2011). Boston, Massachusetts: The MIT Press.