The Future of the Internet is Coordination

Charles Petrie

Stanford University, Stanford, California 94305, USA,
petrie@stanford.edu,
WWW home page: http://www-cdr.stanford.edu/~petrie

Abstract. The key functionality of a Coordinated Internet would be that the Internet actively watches what people do (analogous to search completion on desktops today), correlates these activities, and actively notifies people when and how their current tasks affect and are affected by the activities of other people. Such a Coordinated Internet would provide revolutionary functionality. Some solutions exist, notably in concurrent engineering such as the Redux solution, but foundational research remains to be done in Coordination Engineering.

1 From Web 2.0 to Semantic Web Services

The distinction of the ill-defined Web2.0 technologies from older ones is that in Web2.0, publishing data is more democratic: anyone can publish with no programming. Such Internet (not just web) technologies have contributed to the rise of "Emergent Collectives"[1], [3] that often cause unexpected disruptions in the extant economic models.

Web services offer a next step in this process of dynamic Internet technologies. Web services could be also democratic, which in this case means the service need not be changed by the user. They could also be "just-in-time", which means the user may invoke the service at any time without prior arrangement.

As a consequence, we could have a free market of services: users have a run-time choice among competing services. Especially important for enterprises, this provides flexibility: one could find an alternative service whenever a contingency blocked use of the default service.

Were the democratic and just-in-time principles to apply for all public services on the Internet, they would comprise an open free market of just-in-time services, but development of dynamic open services seems far in the future for various practical reasons[5].

This vision depends critically upon standardized descriptions of services, typically envisioned as the Semantic Web[6], perhaps achieved first by "industrial service parks"[4]. We assume in this paper that the semantics problem will get solved one way or the other, perhaps by linked data[1] for example. We now elaborate the vague reference in [6] about workflows and coordination that semantics would make possible.

[1] http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/

2 The Universe Doesn't Run on Workflow

Many technologies for the dynamic composition of web services have been developed in the last 10 years. If we can discover and consume reusable services with common descriptions, new processes may be dynamically constructed, automatically, to achieve goals[3].

Given a goal, for example, to be reimbursed for travel expense, a new process can be created for this purpose, even just one time for one person and one set of expenses. The processes would take into account all of the currently expressed enterprise policies, such as, for instance, what amounts must be approved by managers and the conditions that qualify to be a manager[2].

Individuals inside companies, and in cross-company projects, could synthesize workflows and other processes *as needed*. Companies could manage such processes by stating explicit policies and constraints that such processes would respect, instead of trusting that programmers would properly interpret these policies, and instead of waiting for new policies to be programmed. We call this *Enterprise Physics*: the universe doesn't run on workflow, so why should enterprises?

Such dynamic processes should, and could, allow for change: contingencies, conflicts, opportunities, policy revisions, and outright failure. This is a further advantage over static workflows.There are supply chain problems, resulting from a needed change in suppliers, that could not be handled by today's workflows [3].

This raises the question of how global dynamic processes could be managed.

3 The Coordinated Internet

Should we achieve the vision of dynamic processes, there is an even more radical future Internet that will then be necessary,: the "Coordinated Internet". This is a vision of a *pro-active* Internet that not only facilitates sharing and collaboration, but which actively coordinates humans, as well as various programs, most notably services.

The key functionality functionality for coordination, beyond passive information sharing, is the notification of changes and their effects to the right people, and programs, at the right time. Example notifications are that a conflict has occurred and who is involved, safe and consistent solution options upon request, thrashing warnings, opportunities for synergy, and that some tasks are no longer necessary.

A Coordinated Internet must actively watch what people do (analogous to search completion on desktops today), correlate these activities, and actively notify people when and how their current tasks affect and are affected by the

[2] "Web services", as in the Dagstuhl definition, `http://tinyurl.com/webservdef`, need not have anything to do with the web but semantics are required even if the implementation is in, say, "apps".

activities of other people. The trick is to do this in a useful way, so that coordination is increased, but people are not annoyed by a "big brother paper clip".

The effects of such future Internet could be world changing, revolutionizing not only how companies are managed, but how any large enterprise is done, especially ad hoc ones, such as global relief efforts for catastrophes[7]. Some of this technology has long existed in the field of concurrent engineering [8].

4 One Mechanism for Coordination: Redux

The Coordinated Internet can be viewed as a the management of a configuration or planning problem worked on by distributed actors. When there are multiple objectives and no single objective function, as there is in the case of large projects with many engineers, the best we can do in terms of optimality is to ensure that no single objective solution can be improved without harming the solution of another: *Pareto optimality* .

The Redux model of design, as implemented in a subset called Redux'[10], implements the use of Pareto optimality within a model of design[9]. Given a conflict (either a constraint violation or a goal block) and a solution (the revision of some design decision), using a justification-based truth maintenance system (JTMS), Redux constructs justifications for revisions resulting from the resolution of conflicts using Dependency-directed backtracking (DDB) that become invalid if the conflict would not obtain if ever there is another way of resolving the conflict, thus enforcing pareto optimality. This justification should not introduce any unsatisfiable circularity into the JTMS network.

Notable characteristics of the Redux model include distinguishing between goals that need to be achieved, and constraints that should not be violated; distinguishing between conditions that affect the optimality of a design decision and its validity; and identifying opportunities, resulting from loss of pareto optimality, as well as conflicts. A set of notifications useful for active coordination is provided in this system. The simplest example is when a subtask has become redundant because the method of achievement of the supertask has changed.

Redux also guarantees no thrashing and identifies safe solutions to conflicts and goal blocks in a distributed environment by management of decision rationales and the justifications produced by conflict resolution.

5 Coordination Engineering

There are other coordination models besides Redux, as well as outstanding issues not solved by Redux. There is a need for much more research in the field of *coordination engineering.* Topics include the proper technologies for detecting and understanding tasks and filtering notifications so that they are more useful to people, than not. Understanding how to combine constraint satisfaction techniques with DDB is a major issue[11].

Better algorithms for underlying justification graph control as well as better representations of committed actions and sunk costs, and providing appropriate transparency of information in supply chains, are known research issues.

Internet implementation questions involve what part of the coordination functionality to embed in what Internet layers. For instance, could the message notification could be handled by smart routers and distributed servers? How should the "watching" function be implemented in browser plug-ins and mobile device apps? In Next-Link[9], we hand-inserted code in standard engineering programs. Is there a more scalable approach?

The Coordinated Internet also has deep issues in common with Internet of Things, such authorization to change descriptions of products and services.

The Coordinated Internet provides a rich new topic of research for computer science as well as the potential to radically improves mankind's ability to manage complex projects.

References

[1] Petrie, C.: Emergent Collectives for Work and Play. *Socite' de Strategie, AGIR Revue Generale de Stratagie*, Nos. 20:21, Jan (2005); `http://www-cdr.stanford.edu/~petrie/revue`.

[3] Petrie, C.: Plenty of Room Outside the Firm. *IEEE Internet Computing*, Peering, , 92–96, Jan-Feb (2010); `http://www-cdr.stanford.edu/~petrie/online/peer2peer/vision2010.pdf`.

[3] Petrie, C.: Planning Process Instances with Web Services. *Proc. ICEIS*, May (2009); `http://logic.stanford.edu/sharing/papers/IVIS09-serviceplanning.pdf`.

[4] Petrie, C., Bussler, C.: The Myth of Open Web Services - The Rise of the Service Parks. *IEEE Internet Computing*, Peering, 80–82 May/Jun (2008); `http://www-cdr.stanford.edu/~petrie/online/peer2peer/serviceparks.pdf.`

[5] Petrie, C.: Practical Web Services. *IEEE Internet Computing*, Peering, , 93–96, N0v-Dec (2009); `http://www-cdr.stanford.edu/~petrie/online/peer2peer/practicalws.pdf`.

[6] Berners-Lee, T., Hendler, J., and Lassila, O.: The Semantic Web. *Scientific American* vol. 184, no. 5 pp. 34–43 May (2001).

[7] Petrie, C.: Collective Work,. *IEEE Internet Computing*, Peering, 80–82 Mar-Apr (2008); `http://www-cdr.stanford.edu/~petrie/online/peer2peer/collectivework.pdf`.

[8] Petrie, C., Goldmann, S., and Raquet, A. : Agent-Based Project Management. *Lecture Notes in AI 1600*, Springer-Verlag, (1999); `http://www-cdr.stanford.edu/ProcessLink/papers/DPM/dpm.html`.

[9] Petrie, C., Webster, T., and Cutkosky, M.: Using Pareto Optimality to Coordinate Distributed Agents. *AIEDAM* **9**, 269–281 (1995); `http://www-cdr.stanford.edu/NextLink/papers/pareto/pareto.html`.

[10] Petrie, C.: The Redux' Server. *Proc. Internat. Conf. on Intelligent and Cooperative Information Systems (ICICIS)*, Rotterdam, May, (1993); `http://www-cdr.stanford.edu/ProcessLink/papers/redux-prime.pdf`.

[11] Petrie, C., Jeon, H., and Cutkosky, M. : Combining Constraint Propagation and Backtracking for Distributed Engineering. *Proc. AAAI'97 Workshop on Constraints and Agents*, AAAI Press, Technical Report WS-97-05, August (1997).