

# Process Detection in Homeland Security and Defense Applications

George Cybenko<sup>a</sup> and Vincent Berk<sup>a</sup>

<sup>a</sup>Thayer School of Engineering and Institute for Security Technology Studies, Dartmouth College, Hanover NH 03755 USA

## ABSTRACT

Process detection is a fundamental problem arising in a variety of homeland security, national defense and commercial applications, including network security, sensor network data fusion, dynamic social network analysis and video tracking of kinematic objects. Our approach to process detection is based on a generic algorithmic approach called Process Query Systems which has been developed at Dartmouth over the past 3 years. This paper surveys the general area of process detection, its applications and recent progress made in various implementations.

**Keywords:** Network security, social network analysis, sensor networks, process detection.

## 1. INTRODUCTION

Many problems of current interest to homeland security and national defense involve the detection and identification of multiple dynamical processes using noisy streams of data collected by a variety of sensors. Table 1 below summarizes some of the application areas together with the dynamical processes and data sources that arise in them.

Environment	Processes	States	Observables
Computer Network Attacks	Host and Network Behaviors	Normal, Scanned, Infected, Failed...	Snort alerts, host-based logs, etc
Server Farms	Server Applications	Normal, Degraded, Failed, Recovered	Performance measures, snort, IDS alerts
National Border	Moving Objects	Position + Velocity	Video, IR images, acoustic, seismic
Geographic Region	Airborne agent diffusion and drift	Releases at times T, locations L	Sensor detection of airborne agent
Identity Theft and Management	Consumer, bank, attacker activities	Normal, phished, exploited, ...	Credit reports, web postings, breaches
Social Networks	Business and social activity	Stages of the activity	Communications, transactions, etc

**Figure 1: Process Detection in Various Application Areas.**

Although these problems appear to be somewhat different superficially, we have developed a mathematical and algorithmic framework in which they are abstractly the same. Specifically, each problem involves detecting *processes* which are defined by states, state transitions and observables related to the states which are typically hidden. Table 1 enumerates these ingredients for the various problems listed. The PQS approach has been applied to a wide variety of problems already but much work remains to be done.<sup>1-16</sup> In this paper, we briefly describe the PQS modeling framework, the PQS algorithmic foundation and several select application areas. Readers are referred to the referenced papers for more details. A recent comprehensive overview of PQS and an implementation can be found in a recent Ph.D. thesis.<sup>17</sup>

---

Further author information:

George Cybenko: E-mail: gvc@dartmouth.edu, Telephone: 1 603 646 3843

Vincent Berk: E-mail: vberk@ists.dartmouth.edu, Telephone: 1 603 646 2230.

## 2. PROCESS QUERY SYSTEM MODELING

We believe that PQS is widely applicable to problem domains in which the challenge is to detect dynamical processes. By a process, we mean a classical state-space based system model that has *states* in the classical sense that the state largely captures the history of the process to the extent that the future evolution of the system depends on the current state and possibly stochastic influences. A process has *dynamics*, namely some mechanism which determines either stochastically, deterministically or nondeterministically what subsequent states are possible. Finally, a process' states generate *observables* that are typically ambiguously related to the states of the process.

The most challenging problems involve processes with *hidden* states; that is, the states are not directly observable but must be inferred from the sequence of observations witnessed. Such problems arise in adversarial situations. An environment consists of processes which evolve over time but which are not fully visible to an observer. In an adversarial setting, that lack of transparency is deliberate. For example, an attack on a computer network consists of several stages or states as we prefer to model this situation. The attack's true states are perhaps abstractions or only known to the attacker. In fact, there is much value added to the attacker if the states of the attack are hidden for as long as possible to allow as deep a penetration and as long a persistence as possible.

Another complication of many current problems in homeland security and national defense is the fact that multiple processes are typically present in the environment. For example, in a computer network under attack, many benign or normal processes are typically instantiated at any given time. Such normal processes include authorized applications communicating, web transactions, file transfers and so on. Accordingly, the process detection problem also typically involves the resolution of which processes are present and what states those processes are in.

From the above discussion, it is clear that processes description formalisms that are suitable include Hidden Markov Models (HMM's), linear and nonlinear state space systems, Petri nets and nondeterministic automata. See our previous work<sup>2</sup> for a discussion of the modeling formalisms in more detail. We have had considerable experience with HMM, nondeterministic automata and state space kinematic-type systems. Details of those models and their applications in various domains can be found in the referenced papers.

To date, the mechanism by which we model an environment has been limited to either first principles or expert judgement. We have not attempted any formal machine learning techniques for extracting process descriptions automatically from data, although this problem is on our long term research agenda.

## 3. PROCESS QUERY SYSTEM ALGORITHMICS

Details of the specific algorithms we have implemented can be found in previous published works.<sup>2,17</sup> Algorithmically, a process query system implementation shares many ingredients conceptually with multi-target tracking algorithms but in a broader domain of process models. Classical multi-target tracking has been based on radar and sonar tracking of multiple objects whose dynamics are relatively simple kinematic models.

Loosely speaking, a PQS implementation must include: methods for assigning observations to one of the process models hypothesized for the environment; methods for scoring those assignments; methods for generating a multitude of possible assignments (called hypotheses) if more than one possible assignment is possible and; a technique for managing the hypotheses so that they do not grow exponentially in number.

Additionally, several applications that we have modeled using the PQS framework naturally benefit from a hierarchical modeling approach. Namely, the hypothesized processes are themselves treated as observables for another level of PQS processing, involving higher level models which describe formations or correlations between processes that have been detected. This has been the case in our work on detecting complex cyberattacks for example.<sup>13</sup> This hierarchical detection capability fits into the JDL fusion hierarchy nicely.

We have developed software implementations of a *Process Query System* (PQS) that can effectively solve these problems if instantiated appropriately.<sup>2,17</sup> In certain respects, a Process Query System implementation is like a Database Management System (DBMS) in that the system provides a generic framework on which to build application programs of interest to end users.

For example, accounting, payroll, transactions processing and purchasing applications can be built from one and the same database software package. The database software provides powerful abstractions for file management, record retrieval, and record manipulation which are efficiently implemented through function calls to the database library. Our PQS implementations are relatively similar in this respect. We believe that effective solutions to a variety of different appearing problems can be implemented through the generic functions that PQS provides.

#### 4. PROCESS QUERY SYSTEM APPLICATIONS

Over the past two years, we have applied the PQS framework to problems involving computer network security,<sup>6,7,13</sup> autonomous computing,<sup>8,11</sup> insider threats,<sup>5</sup> tracking using sensor networks and UAVs,<sup>1,9,15</sup> social network analysis<sup>14</sup> and motion tracking using infrared video.<sup>16</sup>

Details and additional references to these applications can be found at the website [www.pqsnet.net](http://www.pqsnet.net).

We believe that one remarkable aspect of this corpus of work is the breadth and relative performance achievable by using a common algorithmic framework across several different application areas that superficially appear to be quite different from each other.

#### 5. THE FUTURE OF PROCESS QUERY SYSTEMS

We believe that there are many problems amenable to the PQS modeling and algorithmic formalism. Our goals are to continue development of the basic concepts and to apply them to a larger collection of problem domains, while continuing the refinement of the applications we have already attempted.

Several analytic problems remain outstanding. For example, as mentioned above, mechanisms for automatically generating process models will be necessary in some problem domains for which insights and first principles will be hard to extract, but in which large volumes of data are available. Another technical challenge of PQS that we have only superficially explored is the hypothesis management problem, which also arises in multiple target tracking using the multiple hypothesis tracking approach. Specifically, techniques for maintaining reasonably good but diverse hypotheses is needed but there has been little broadly applicable work done in that area so far.

We are continuing to explore these algorithmic and analytic areas.

#### 6. ACKNOWLEDGEMENTS

Research described in this paper was partially supported by DHS/ODP Grant 2000-DT-CX-K001, NGIA contract HM1582-05-1-2033, ORNL UT-Batelle (DOE) Grant 4000047683 and DTO/ARDA P2INGS Award F30602-03-C-0248. All opinions expressed are solely those of the authors and not the sponsoring organizations.

#### REFERENCES

1. V. Crespi *et al.*, "Process query systems for surveillance and awareness," in *7th World Multiconference on Systemics, Cybernetics and Informatics*, SCI2003, (Orlando, FL), July 27-39.
2. G. Cybenko, V. H. Berk, V. Crespi, R. S. Gray, and G. Jiang, "An overview of process query systems," in *Proceedings of the SPIE Vol. 5403, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, SPIE, (Orlando, FL), 2004.
3. G. T. Nofsinger and K. W. Smith, "Plume source detection using a process query system," in *Proceedings of the SPIE Vol. 5416 Chemical and Biological Sensing V*, SPIE, (Orlando, FL), April 2004.
4. G. Nofsinger and G. Cybenko, "Distributed chemical plume process detection," in *Proceedings of IEEE MILCOM*, (Atlantic City, NJ), 2005.
5. P. Thompson, "Weak models for insider threat detection," in *Proceedings of the SPIE Vol. 5403, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, (Orlando, Florida), April 2004.
6. V. Berk and N. Fox, "Process query systems for network security monitoring," in *Proceedings of the SPIE Vol. 5403, Defense and Security Symposium*, (Orlando, Florida), March/April 2005.

7. V. Berk, A. Giani, and G. Cybenko, "Covert channel detection using process query systems," in *Proceedings of FLOCON - CERT, 2nd Annual Workshop on Flow Analysis*, (Pittsburgh, PA), September 2005.
8. G. Cybenko, V. Berk, and C. Roblee, "Large-scale autonomic server monitoring using process query systems," in *Proceedings of the SPIE Vol. 5403, Defense and Security Symposium*, (Orlando, Florida), March/April 2005.
9. V. Crespi, W. Chung, and A. B. Jordan, "Decentralized sensing and tracking for UAV scheduling," in *Proceedings of the SPIE Vol. 5403, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, (Orlando, Florida), April 2004.
10. D. Hernando and V. Crespi, "Sampling theory for process detection with applications to surveillance and tracking," in *Proceedings of the SPIE Vol. 5403, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, (Orlando, Florida), April 2004.
11. C. Roblee, V. Berk, and G. Cybenko, "Implementing large-scale autonomic server monitoring using process query systems," in *Proceedings of 2nd IEEE International Conference on Autonomic Computing (ICAC-05)*, (Seattle, WA), June 2005.
12. A. Giani, V. Berk, and G. Cybenko, "Data exfiltration and covert channels," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, (Orlando, FL), April 2006.
13. I. DeSouza *et al.*, "Detection of complex cyber attacks," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, (Orlando, FL), April 2006.
14. W. Chung *et al.*, "Dynamics of process-based social networks," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, (Orlando, FL), April 2006.
15. V. Crespi, G. Cybenko, and Y. Sheng, "Tracking in complex situations and environments," in *Unattended Ground, Sea, and Air Sensor Technologies and Applications VIII Conference*, (Orlando, FL), April 2006.
16. V. Berk *et al.*, "Target tracking and localization using infrared video imagery," in *Unattended Ground, Sea, and Air Sensor Technologies and Applications VIII Conference*, (Orlando, FL), April 2006.
17. V. Berk, "Process query systems." Ph.D. Thesis, Leiden University, Netherlands, January 2006.