

ANDREA: Scalable Command-and-control of Distributed Applications

Jonathan C. Hill, John C. Knight
Department of Computer Science, University of Virginia
{jch8f, jck}@cs.virginia.edu

ANDREA is a command-and-control service architecture for very large, distributed applications. Command-and-control systems are appealing because they apply directed action, a strategy familiar in the form of imperative programming. Tenability for this approach at very large scales, however, requires that controllers relax their system-state-knowledge. ANDREA supports this requirement with Site-Select Command and Intention Council; efficient distributed mechanisms enhancing programmable command-and-control. The resulting architecture supports controlled emergence, a ‘middle-ground’ between imperative control and emergent group behavior.

1. Introducing ANDREA

Command-driven systems traditionally apply sensors, controllers, and actuators within control loops. Today’s variants typically apply controller agents. Agents are autonomous, utilizing knowledge of system state and other controller agents in social organizations. The complexity of maintaining imperative, central-command driven approaches within social models induces poor scaling for command-and-control programming. As a result, researchers are increasingly studying application of self-organized emergent group behavior. Yet there is still a desire to interface these models with human cognitive methods, including imperative command-and-control.

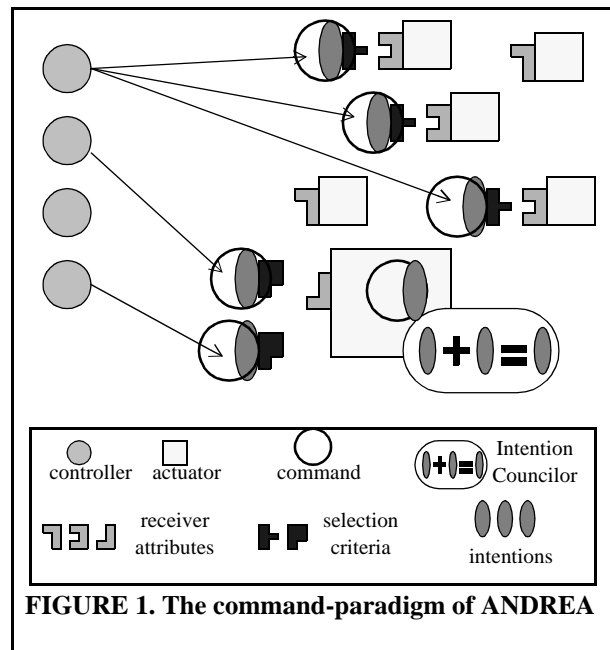
ANDREA [2] is a command-and-control architecture supplying useful ‘middle-ground’ between imperative command and a class of emergent behaviors. This is supported through two service/algorithms:

- **Site-Select Messaging:** a distributed and efficient means of one-to-many communication, allowing controllers to specify the run-time attributes and conditions desired of command recipients, then efficiently delivering commands to all qualifying actuators, and
- **Intention Council:** a distributed means of efficiently and meaningfully resolving conflicts between converging actions.

The two techniques, respectively, handle divergence and convergence of commands in very large-scale systems.

Figure 1 illustrates Site-Select Messaging in a ‘puzzle’ analogy. (Antibody and chemical messenger analogies also apply.) Lightly-shaded ‘puzzle-pieces’ attached to each receiver (actuator) represent receiver attributes. A command message’s selection expression is depicted as a darkly-shaded ‘puzzle-piece.’ Commands travel to all ‘matching puzzle pieces.’

Intention Council is shown in the figure by the analogy of an ‘equation’ at a receiver. Intention Councilors are agents observing converging commands, deciding which commands to allow to execute, which to stall, and which to throw away.



2. Model-Enabled

ANDREA is an intermediate systems layer. It applies run-time binding algorithms to separate the roles of controller, actuation relater, and conflict resolver as respectively handled by controllers, Site-Select Messaging, and Intention Councilors. Both Site-Select Messaging and Intention Council introduce programmable models.

2.1 Receiver Model and Site-Select Messaging

Site-Select Messaging utilizes a programmable receiver model. The receiver model consists of typed, named attributes with independent values at each receiver site. A language of Boolean selection expressions (similar to relational database selection expressions) is available to messages for selecting receivers based on their attribute values.

Commands propagate efficiently through publish/subscribe messaging, arriving at every (and only) receiver(s) matching the command's requirements [1]. Siena's peer-to-peer publish/subscribe performs this routing with distributed efficiency [3]. Content-driven publish-subscribe [3] allows a wide range of selection expressions [4].

Consider the utility of this command approach. Attributes can mark semi-static attributes such as receiver security levels, regional location, hierarchical organization, and assigned roles in survivability plans. Other attributes can describe more frequently variable state including activity levels, security alerts, current users, and usage statistics. Commands can be targeted to actuator conditions.

2.2 Intention Model and Intention Counseling

Intention Council allows controller societies to lessen their inter-coordination complexity through a system-wide, programmed model. Social contract is applied at the receiver-end of command, governed by controlled interaction of converging commands. These interactions utilize intention data stated within commands. Stated intentions are drawn from a formal domain, programmed to represent what a command plans to do and why. The programmed intention domain maintains a total order, so that standard resource reservation and scheduling can be applied to commands from within ANDREA's scheduling algorithms.

Each receiver maintains an Intention Council. This is a collection of Intention Councilors deciding which received commands are acceptable. Intention Councilors are defined as machines observing past and present commands. Intention Councilors apply programmed rules to delay, throw out, or allow commands to execute based on their intentions. In essence, Intention Councilors determining command responsiveness at the receiver-end.

3. Controlled-Emergence and Distributed State

Together, Site-Select and Intention Council form run-time binding over otherwise explicit programming. Just as virtual functions and run-time method binding are power-

ful and useful mechanisms in object-oriented programming, Site-Select and Intention Council allow powerful run-time binding within control programming of large distributed systems.

The extent to which control properties are run-time determined or explicitly specified within control programs is a function of the programming of the Receiver and Intention Models as well as controllers. This allows *controlled-emergence* of properties. Systems can organize and respond based on run-time conditions, enacting a run-time emergent response, based on the imperative commands of controllers. Possible applications include 'growing' system installation topology. Another interesting possibility is the application of Site-Select and Intention Council in the self-organization of controller and Intention Councilor societies.

ANDREA is not a centralized service. Furthermore, ANDREA promotes distribution of controller and actuator technology. It supports societies in which controllers and actuators apply expert action independent of their social restrictions. Programming of Resource Models and Intention Models formulates the resulting social order.

4. Future Work

Our experiments will attempt to measure the scaling and efficiency of the ANDREA architecture. We intend to experiment to determine the effectiveness and programmability of controlled systems for survivability domains including application of controlled-emergence.

5. References

- [1] Gerkey, Brian P. Mataric, Maja J. 'Murdoch.' *Proceedings of the fourth international conference on Autonomous agents June 2000*. ACM Press.
- [2] Knight, J. Heimbigner, D. Wolf, A. Carzaniga, A. Hill, J. Devanbu, P. Gertz, M. 'The Willow Survivability Architecture.'
- [3] Carzaniga, A. Rosenblum, D. S. Wolf, A.L. 'Achieving Expressiveness and Scalability in an Internet-Scale Event Notification Service'. *Nineteenth ACM Symposium on Principles of Distributed Computing (PODC2000), Portland OR, July, 2000*. ACM Press.
- [4] Hill, J. C. 'Site-Select Messaging.' *University of Virginia, Technical Report #CS-2002-06*