UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ACTIVISION BLIZZARD, INC.,
ELECTRONIC ARTS INC.,
TAKE-TWO INTERACTIVE SOFTWARE, INC.,
2K SPORTS, INC.,
ROCKSTAR GAMES, INC., and
BUNGIE, INC.,
Petitioner,
v.

ACCELERATION BAY, LLC,
Patent Owner.

Case IPR2015-01951[1]
Patent 6,714,966

DECLARATION OF DR. FRED B. HOLT IN SUPPORT OF PATENT OWNER'S RESPONSE

[1] Bungie, Inc., who filed a Petition in IPR2016-00935, has been joined as a petitioner in this proceeding.

I, Fred B. Holt, Ph.D., declare as follows:

1. I am over the age of majority and make this declaration of my own personal knowledge.

2. I was employed at Boeing, Inc. ("Boeing") initially in 1984, from 1986 to 1992 and from 1996 to January 2002 as a research scientist and mathematics modeler. While at Boeing, I was responsible for inventing and improving solutions to Boeing-internal technology challenges.

3. I am a co-inventor of the "SWAN: Small-World Wide Area Networking" ("SWAN") invention, which is the subject matter of the patented inventions covered by U.S. Patent Nos. 6,829,634 ("the '634 Patent"), 6,701,344 ("the '344 Patent"), 6,920,497 ("the '497 Patent"), 6,910,069 ("the '069 Patent"), 6,732,147 ("the '147 Patent"), and 6,714,966 ("the '966 Patent") (collectively, the "SWAN Patents").

4. In the Fall of 1996, after I completed my Ph.D. in Mathematics at the University of Washington, I returned to the Mathematics & Engineering Analysis Group at the Boeing Company ("Boeing") as a research scientist.

5. Shortly after my return, Virgil Bourassa came to my office to discuss a project he was working on for [redacted] a CAD visualization tool that was used internally at Boeing. At that time, individual engineers generally used [redacted] as a stand-alone application and at times with a one-to-one connection for two-person

collaboration. Virgil informed me that they had asked Virgil to expand the communications for collaboration to allow for three-person collaboration. Virgil knew that as soon as we would deliver a system that allowed for three-person collaboration, there would eventually be a request to keep increasing the number of participants, so we needed to figure out how to solve this beyond a three-person capability. We then talked about the requirements and constraints for this request, and I suggested random regular graphs as a potential solution. We immediately began identifying issues and working on solutions for a real deployment.

6. As Virgil and I described in our Invention Disclosure Form, the typical computer network communications techniques included: (1) fully-connected point-to-point network protocols, (2) client/server middleware, (3) multicasting network protocols, and (4) peer-to-peer middleware.

7. Fully-connected point-to-point networking protocols were disadvantageous because fully-connected network graphs do not scale as the number of participating processes grows. Client/server middleware systems were disadvantageous because the server is a performance bottleneck and a single point of failure. Multicast networking protocols were disadvantageous because multicast traffic at the time was limited to single local-area networks. Peer-to-peer middleware at the time was not suitable for the needs of medium to large-scale collaboration.

8. The lack of technology available that could provide peer-to-peer communications among computer processes across the world with high reliability and low latency reaffirmed that a solution was needed in this area to satisfy an existing need in the market. Therefore, Virgil and I realized that there was a need to create an effective means to allow scalable and reliable sharing of information across multiple processes.

9. Virgil led the effort to develop SWAN. Virgil was responsible for communicating with the [redacted] project group as we worked on SWAN. I proposed using random regular graphs as the communications topology. I also gave an initial explanation about why the degree should be even and at least four.

10. One of the major constraints that Virgil insisted on in the design process was that we had to keep the information local and asynchronous. The desired topology (random regular graphs of even degree) and the method of broadcasting messages were determined at the outset.

11. Within two months, Virgil and I had produced an initial simulation of SWAN, and we had started on a research implementation of SWAN. The simulation showed how the graphs of the communications topology changed as the number of participants grew. We did not use the simulation for very long, since the research implementation very quickly provided more capabilities. The research implementation would launch very simple UNIX processes that would join a

SWAN session. We could launch a few dozen nodes on a single UNIX computer before swamping the resources of this single computer.

12. For joining a SWAN session, the first challenge we had to solve was how to pick n/2 random edges in the existing graph by using only local information. We knew that we wanted to keep the diameter of the resulting graph low, and preferred that the n/2 edges were "far" from each other in the graph. If the edges were all chosen too closely to the initiating node, the graph grew into a sausage-shape with a diameter that was too high. The solution we chose was to initiate n/2 random walks through the existing fabric. The length of the random walks would be proportional to estimates of the current diameter. Since the true diameter was a global parameter, we had to use estimates based on local information. Ideally, the random walks would have a length equal to the square of the diameter. Through experimentation, we settled on a heuristic of a low multiple of the diameter as sufficient.

13. The joining algorithm had to move seamlessly through and out of the "small regime" in which there are fewer than n+1 nodes. If the session is in the small regime, then there is no need to launch the random walks. We developed a solution to determine whether we are in the small regime using only local information, and if so, to introduce a new node into the session in this context.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.