# WITHDRAWAL SHEET

## Ronald Reagan Library

**Collection Name** MATLOCK, JACK: FILES

**Withdrawer** JET 4/11/2005

**File Folder** MATLOCK CHRON (APPROVED/DISAPPROVED) SEPTEMBER 1985 (3/6)

**FOIA** F06-114/3 YARHI-MILO 1106

**Box Number** 11

| ID | Doc Type | Document Description | No of Pages | Doc Date | Restrictions |
|---|---|---|---|---|---|
| 7814 | MEMO | COBB/MATLOCK TO MCFARLANE RE NSC MEETING ON SHEVARDNADZE VISIT--- FRIDAY SEPTEMBER 20, 1985----11:00 AM<br>R 3/8/2011 F2006-114/3 | 2 | 9/19/1985 | B1 |
| 7815 | MEMO | MCFARLANE RE MEETING WITH THE NSC SEPTEMBER 20, 1985<br>R 3/8/2011 F2006-114/3 | 4 | 9/20/1985 | B1 |
| 7816 | MEMO | COBB/MATLOCK TO MCFARLANE RE YOUR TALKING POINTS FOR SHEVARDNADZE NSC MEETING FRIDAY, SEPTEMBER 20, 1985---11:00 AM<br>R 3/8/2011 F2006-114/3 | 1 | 9/19/1985 | B1 |
| 7817 | TALKING POINTS | TALKING POINTS FOR SHEVARDNADZE NSC MEETING<br>R 3/8/2011 F2006-114/3 | 3 | 9/19/1985 | B1 |
| 7818 | AGENDA | AGENDA FOR NSC MEETING FRIDAY SEPTEMBER 20, 1985 11:00 AM<br>R 3/8/2011 F2006-114/3 | 1 | 9/19/1985 | B1 |
| 7819 | MEMO | SAME TEXT AS DOC #7816<br>R 3/8/2011 F2006-114/3 | 1 | 9/19/1985 | B1 |
| 7820 | TALKING POINTS | SAME TEXT AS DOC #7817<br>R 3/8/2011 F2006-114/3 | 3 | ND | B1 |

Freedom of Information Act - [5 U.S.C. 552(b)]

B-1 National security classified information [(b)(1) of the FOIA]
B-2 Release would disclose internal personnel rules and practices of an agency [(b)(2) of the FOIA]
B-3 Release would violate a Federal statute [(b)(3) of the FOIA]
B-4 Release would disclose trade secrets or confidential or financial information [(b)(4) of the FOIA]
B-6 Release would constitute a clearly unwarranted invasion of personal privacy [(b)(6) of the FOIA]
B-7 Release would disclose information compiled for law enforcement purposes [(b)(7) of the FOIA]
B-8 Release would disclose information concerning the regulation of financial institutions [(b)(8) of the FOIA]
B-9 Release would disclose geological or geophysical information concerning wells [(b)(9) of the FOIA]

C. Closed in accordance with restrictions contained in donor's deed of gift.

# WITHDRAWAL SHEET

## Ronald Reagan Library

**Collection Name** MATLOCK, JACK: FILES

**Withdrawer** JET 4/11/2005

**File Folder** MATLOCK CHRON (APPROVED/DISAPPROVED) SEPTEMBER 1985 (3/6)

**FOIA** F06-114/3

**Box Number** 11

YARHI-MILO 1106

| ID | Doc Type | Document Description | No of Pages | Doc Date | Restrictions |
|---|---|---|---|---|---|
| 7821 | AGENDA | SAME TEXT AS DOC #7818<br>***R 3/8/2011 F2006-114/3*** | 1 | 9/19/1985 | B1 |
| 7822 | MEMO | MATLOCK TO POINDEXTER RE MEETING WITH TOM JOHNSON, SATURDAY, SEPTEMBER 21, 11:30AM<br>***R 10/30/2007 NLRRF06-114/3*** | 1 | 9/20/1985 | B1 |
| 7823 | MEMO | SAME TEXT AS DOC #7822<br>***R 10/30/2007 NLRRF06-114/3*** | 1 | 9/20/1985 | B1 |
| 7825 | MEMO | PARNAS TO OFFUT RE SDIO PANEL ON COMPUTING IN SUPPORT OF BATTLE MANAGEMENT<br>***R 6/6/2007 F06-114/3*** | 2 | 6/28/1985 | B1 |
| 7826 | PAPER | WHY SOFTWARE IS UNRELIABLE BY PARNAS<br>***R 6/6/2007 F06-114/3*** | 2 | ND | B1 |
| 7827 | PAPER | WHY THE SDI SOFTWARE SYSTEM WILL BE UNTRUSTWORTHY BY PARNAS<br>***R 6/6/2007 F06-114/3*** | 2 | ND | B1 |
| 7828 | PAPER | WHY CONVENTIONAL SOFTWARE DEVELOPMENT DOES NOT PRODUCE RELIABLE PROGRAMS<br>***R 6/6/2007 F06-114/3*** | 2 | ND | B1 |

**Freedom of Information Act - [5 U.S.C. 552(b)]**

**B-1 National security classified information [(b)(1) of the FOIA]**
**B-2 Release would disclose internal personnel rules and practices of an agency [(b)(2) of the FOIA]**
**B-3 Release would violate a Federal statute [(b)(3) of the FOIA]**
**B-4 Release would disclose trade secrets or confidential or financial information [(b)(4) of the FOIA]**
**B-6 Release would constitute a clearly unwarranted invasion of personal privacy [(b)(6) of the FOIA]**
**B-7 Release would disclose information compiled for law enforcement purposes [(b)(7) of the FOIA]**
**B-8 Release would disclose information concerning the regulation of financial institutions [(b)(8) of the FOIA]**
**B-9 Release would disclose geological or geophysical information concerning wells [(b)(9) of the FOIA]**

**C. Closed in accordance with restrictions contained in donor's deed of gift.**

# WITHDRAWAL SHEET

## Ronald Reagan Library

**Collection Name** MATLOCK, JACK: FILES

**Withdrawer** JET 4/11/2005

**File Folder** MATLOCK CHRON (APPROVED/DISAPPROVED) SEPTEMBER 1985 (3/6)

**FOIA** F06-114/3

**Box Number** 11

YARHI-MILO 1106

| ID | Doc Type | Document Description | No of Pages | Doc Date | Restrictions |
|---|---|---|---|---|---|
| 7829 | PAPER | THE LIMITS OF SOFTWARE ENGINEERING METHODS BY PARNAS<br>***R 6/6/2007 F06-114/3*** | 3 | ND | B1 |
| 7830 | PAPER | ARTICIFICAL INTELLIGENCE AND SDI BY PARNAS<br>***R 6/6/2007 F06-114/3*** | 2 | ND | B1 |
| 7831 | PAPER | CAN AUTOMATIC PROGRAMMING SOLVE THE SDI SOFTWARE PROBLEM BY PARNAS<br>***R 6/6/2007 F06-114/3*** | 2 | ND | B1 |
| 7832 | PAPER | CAN PROGRAM VERIFICATION MAKE THE SDI SOFTWARE RELIABLE BY PARNAS<br>***R 6/6/2007 F06-114/3*** | 2 | ND | B1 |
| 7833 | PAPER | IS SDIO AN EFFICIENT WAY TO FUND WORTHWHILE RESEARCH BY PARNAS<br>***R 6/6/2007 F06-114/3*** | 2 | ND | B1 |
| 7824 | MEMO | MATLOCK TO MCFARLANE RE POSSIBLE DEAL TO FREE SHCHARANSKY<br>***R 10/30/2007 NLRRF06-114/3*** | 1 | 9/23/1985 | B1 |

**Freedom of Information Act - [5 U.S.C. 552(b)]**

**B-1 National security classified information [(b)(1) of the FOIA]**
**B-2 Release would disclose internal personnel rules and practices of an agency [(b)(2) of the FOIA]**
**B-3 Release would violate a Federal statute [(b)(3) of the FOIA]**
**B-4 Release would disclose trade secrets or confidential or financial information [(b)(4) of the FOIA]**
**B-6 Release would constitute a clearly unwarranted invasion of personal privacy [(b)(6) of the FOIA]**
**B-7 Release would disclose information compiled for law enforcement purposes [(b)(7) of the FOIA]**
**B-8 Release would disclose information concerning the regulation of financial institutions [(b)(8) of the FOIA]**
**B-9 Release would disclose geological or geophysical information concerning wells [(b)(9) of the FOIA]**

**C. Closed in accordance with restrictions contained in donor's deed of gift.**

NSC/S PROFILE UNCLASSIFIED ID 8507176

RECEIVED 06 SEP 85 15

TO MCFARLANE FROM PRICE, CHARLES DOCDATE 26 AUG 85

MATLOCK 11 SEP 85

KEYWORDS: USSR NUC GORBACHEV, MIKHAIL S

MP

SUBJECT: LTR TO MCFARLANE FM WORLD FEDERALIST ASSOC

---

ACTION: PREPARE MEMO FOR MCFARLANE DUE: 21 SEP 85 STATUS X FILES WH

---

| FOR ACTION | FOR CONCURRENCE | | FOR INFO |
|---|---|---|---|
| MATLOCK | SESTANOVICH | MANDEL | HALL |
| | | | THOMPSON |

COMMENTS

REF# LOG 8507272 NSCIFID ( DR )

---

| ACTION OFFICER (S) | ASSIGNED | ACTION REQUIRED | DUE | COPIES TO |
|---|---|---|---|---|
| Rm | 9/17 | Ltr Ltg | 9/21 | |
| C | 9/19 | McFarlane sgd | | jm, w/att |
| | | | | |
| | | | | |
| | | | | |

DISPATCH ✓ W/ATTCH FILE ____ (C)

2

THE WHITE HOUSE

WASHINGTON

September 19, 1985

Dear Mr. Price:

Thank you for your letter of August 26 and your suggestion that I meet with the members of the Soviet Peace Committee who will be visiting you.

Unfortunately, my schedule will not permit such a meeting during the October visit of your delegation. I would like you to know that I share your goal of world peace, and that we are giving all Soviet statements serious attention.

Please give my best regards to the delegation.

Sincerely,

Robert C. McFarlane

Charles C. Price
Chairman, Executive Committee
World Federalist Association
418 7th Street, S.E.
Washington, D.C. 20003

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

7176

ACTION

September 11, 1985

MEMORANDUM FOR ROBERT C. McFARLANE

FROM: JACK F. MATLOCK SIGNED

SUBJECT: Response to Charles Price

Charles Price, Chairman of the Executive Committee, World Federalist Association, has written you asking for a meeting with members of his organization during October. The organization is a peace group with extensive ties with the Soviet Union. The delegation will be led by Yuri Zhukov, one of the more notorious Soviet propagandists. Needless to say, it would not be at all appropriate or desirable for you to meet with any members of this organization.

RECOMMENDATION

That you sign the thank you letter to Mr. Price for his invitation and concern but inform him that your busy schedule will not permit such a meeting.

Approve ✓ Disapprove _____

Attachments

Tab I Ltr for Signature

Tab II Incoming Letter

cc: Doug Doan

# World Federalist Association

418 7th STREET, S.E., WASHINGTON, DC 20003 (202) 546-3950

Seeking the abolition of war through just and enforceable world laws


9/3 4



August 26, 1985

Honorable Robert MacFarlane
National Security Advisor
The White House
Washington, DC

Dear Mr. MacFarlane:

A year ago a delegation from the World Federalist Association was invited to Moscow by Yuri Zhukov, President, Soviet Peace Committee, for nine days in Moscow at their expense. We met with leaders from nine different groups for very friendly and useful discussions on alternatives to war.

We have invited a delegation from the Soviet Peace Committee to come to Washington this fall. A delegation of four, led by Yuri Zhukov, will be our guests in Washington from October 17-23. Should you or your aides wish to meet with them, we will be pleased to arrange a time suitable to you.

Enclosed is a memo from Academic Leaders for Alternatives to War, addressed to President Reagan and Chairman Gorbachev. It would seem to us highly desirable to explore what Chairman Gorbachev really has in mind in his comments quoted therein. Perhaps an informal opportunity to discuss this briefly with Yuri Zhukov, a member of the Supreme Soviet, might be useful.

I will call your office in early September to see whether we can arrange a meeting.

Best wishes,

Charles C. Price
Chairman, Executive Committee

CCP:es
Enclosure
cc: Walter Hoffmann
Edward Rawson
John Holden
Mr. Jack F. Matlock

Academic Leaders for

# Alternatives to War

15 Dogwood Lane, Swarthmore, PA 19081, 215-543-8947

5

28 May 1985

to: President Ronald Reagan
Chairman Mikhail Gorbachev
re: Alternatives to War

Our group has supported the principles of the 1961 McCloy-Zorin "Joint Statement of Agreed Principles for Disarmament Negotiations" as a statement offering hope for an effective and acceptable alternative to war. This agreement calls for general and complete disarmament and a United Nations effective in peaceful conflict resolution, with a Peace Force to maintain order in a disarmed world. It also calls for an International Disarmament Organization with veto-free access anywhere in the world as required to assure compliance with the disarment agreement. We are pleased that Vice-President George Bush and Thomas Simons, Director, Office of Soviet Union Affairs, Department of State, have each written to us recently in support of the McCloy-Zorin goals.

President Reagan, you have called for a world free of nuclear arms and have stated in your 1983 address to the United Natios that "our goals of peace and justice require the rule of law in international affairs." As a former World Federalist Advisory Board member, you will be interested to know that a World Federalist delegation invited to Moscow in November 1984 repeated this message to the nine groups of Soviets with whom we met.

Chairman Gorbachev, in your 8 May 1985 speech commemorating the end of World War II, you stated "Today, on the day of the anniversary memorable to all of us, I should like to repeat once more: The Soviet Union resolutely comes out for a world without wars, for a world without weapons. We state again and again that the outcome of the historical competition between the two systems cannot be resolved by military means."

You also stated that "We firmly believe that the process of detente should be revived. This does not mean, however, a simple return to what was achieved in the 70's. It is necessary to strive for something mucg greater. From our point of view, detente is not the ultimate aim of policy. It is needed, but only as a transitional stage from a world cluttered with arms, to a reliable and all-embracing system of international security."

It would seem that one urgent purpose of a summit meeting should be for the two of you to agree on this broad vision of a just, peaceful and disarmed world. Furthermore, it would be wise to go betond these statements by seeking agreement on establishing an ongoing international confrence program to define more clearly than did the McCloy-Zorin agreement, a mutually-acceptable goal and agreed major steps to achieve it, including effective and acceptable international institutions necessary to maintain a just and peaceful world order for our common welfare and security.

For the future of humanity, there is no more urgent need than to end the threat of nuclear war. There will be no better time to start than now!

Charles C. Price

7814 JMC 1

SYSTEM II
90939
Add-on

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

~~SECRET~~

September 19, 1985

ACTION

MEMORANDUM FOR ROBERT C. MCFARLANE

FROM: TYRUS W. COBB/JACK F. MATLOCK

SUBJECT: NSC Meeting on the Shevardnadze Visit - Friday September 20,1985 -- 11:00 a.m.

Attached at Tab I is a memorandum from you to the President providing background on the NSC meeting scheduled for September 20, 1985, at 11:00 a.m. in the Cabinet Room. The memorandum includes a proposed agenda at Tab A. The list of participants and the talking points for your use will be provided in a separate package.

Per your instruction this NSC meeting is designed to serve as a preliminary review of the major issues between the United States and the Soviet Union in the bilateral, regional and human rights areas. Arms control philosophy and issues will not be covered specifically, although broader security concerns will certainly be reviewed. We see this session as an opportunity for the senior members of this Administration to discuss candidly with the President our broad approach to our policy toward the Soviet Union and to stimulate discussion on the contrasting Soviet and American objectives for the meeting in Geneva between the President and the General Secretary.

We would anticipate that you would begin this session with an introduction highlighting to the President that this session is intended to serve as a strategic overview of our broad policy toward the Soviet Union and to examine the major issues on our bilateral, regional and human rights agenda. In addition, we recommend that you review for the President our public diplomacy strategy and highlight the key events between now and November 18 that present opportunities for us to convey our position convincingly to the American and Allied publics -- and to the Soviet leadership. You might then turn to Secretary Shultz who will follow with a review of the key Soviet-American issues in the three areas and delineate our objectives for the meeting with Gorbachev. Bill Casey is prepared to follow with a five minute presentation on Soviet objectives, followed by Cap who will treat Soviet defense concerns generally.

~~SECRET~~
Declassify on: OADR

8

Ron Lehman, Bob Linhard, and Johnathan Miller concur.

RECOMMENDATION:

That you sign and forward the memorandum at Tab I.

Approve _____ Disapprove _____

Attachments:

Tab I Memorandum to the President

Tab A Proposed Agenda

Tab B List of Participants

7815

9

~~SECRET~~

SYSTEM II
90939
Add-on

THE WHITE HOUSE

WASHINGTON

MEETING WITH THE NATIONAL SECURITY COUNCIL

DATE: September 20, 1985
LOCATION: Situation Room
TIME: 11:00 a.m. - 12:00 noon

FROM: ROBERT C. MCFARLANE

I. PURPOSE

To review our agenda for the Shevardnadze visit, survey broad Soviet strategic objectives for the Geneva meeting between you and the General Secretary and to discuss our long-range objectives for managing the Soviet-American relationship.

II. BACKGROUND

Your upcoming meeting with Soviet Foreign Minister Shevardnadze provides an opportunity for us to review the broad outlines of our policy towards the Soviet Union, the USSR's strategic objectives, and the Soviet and American "game plans" leading to the November meeting. Today's session will focus on the bilateral, regional and human rights issues between the U.S. and the USSR. We will hold the arms control issues for future sessions.

The Soviet game plan is becoming increasingly clear. They are seeking to create the public impression that they have exerted every possible effort in order to achieve a breakthrough in Geneva. We will want to be receptive to any serious Soviet proposal in order to lay the basis for further steps in our dialogue on the range of bilateral, regional and human rights issues.

-- REGIONAL ISSUES

While public attention has focused on the arms control aspects of our relationship, the Soviet use of force outside Soviet borders lies at the root of our problems. We are particularly concerned with the Soviet tendency to employ military force, directly or through surrogates, in their conduct of foreign policy. In the 1970's our efforts to develop an understanding with the USSR was severely impaired by Moscow's unrelenting pursuit of unilateral advantage.

~~SECRET~~
Declassify on: OADR

10

We will look to our own strength, as well as closer cooperation with our Allies and friends, to defend our interests. We will make it clear to Shevardnadze that we will continue to pursue such policy as necessary -- in Central America, the Middle East, Africa or elsewhere. Further we will not foreswear the right to lend assistance to democratic elements when they appeal to us to resist aggression. At the same time we are seeking to expand our dialogue with the Soviet Union on regional issues. As you know, this year we have had discussions on the Middle East, Southern Africa, Afghanistan and Asia.

-- BILATERAL ISSUES

It may be possible to complete several negotiations on issues such as exchanges and consulates in time for our November meeting. If the Soviets are not forthcoming on these issues, we are prepared to continue our discussions in the future. The important thing is to get agreements which can stand up to the test of time and are firmly grounded on each side's interests.

We told the Soviets we are prepared to discuss resumption of bilateral air service (which is very important to them), once agreement is reached on North Pacific safety measures and how to achieve a balance of economic benefits for American carriers serving the USSR. The Soviets in turn have tied the opening of Consulates and an exchanges agreement to the resumption of Aeroflot service to the U.S.

We may wish to consider some more ambitious proposals in the area of contacts, exchanges and reciprocal access to each other's media, since such steps would undermine the Soviet monopoly of information made available to its own people.

-- HUMAN RIGHTS

The Human Rights situation in the Soviet Union has, if anything, deteriorated since Gorbachev took power. Andrei Sakharov and his wife remain isolated and conditions for well known dissidents such as Shcharanskiy, Orlov, and Begun have deteriorated. In addition, several spouses of American citizens continue to be refused permission to emigrate.

Recent Gorbachev comments and an article by KGB Head Chebrikov call for a noncompromising hardline on internal dissent. The Soviets now respond to our criticism of their human rights performance with aggressive counter attacks on economic and social conditions in the west. Gorbachev advisor Yuri Arbatov recently made it clear to the Vice President that any serious discussion of human rights could spell failure for the summit.

We want to emphasize to Shevardnadze that we consider human rights an integral part of our relationship and an area where the Soviets can do much to improve relations at a low cost to themselves. We may want to tie an improvement with trade relations to moves in the human rights area. Some of these points are best made privately with Soviet interlocutors, to give them the opportunity to adjust their practices without being seen as backing down under U.S. pressure.

III. PARTICIPANTS

To be provided separately.

IV. PRESS PLAN

Photo opportunity in the Cabinet room prior to the meeting.

V. SEQUENCE OF EVENTS

I will introduce the subject highlighting the main issues, followed by George who will provide a review of the key Soviet-American issues. Bill Casey and Cap Weinberger will have 5 minutes each to discuss Soviet strategic objectives in general terms, followed by a 30-minute discussion and concluding remarks.

Prepared by:
Tyrus W. Cobb/Jack F. Matlock

Attachment
Tab A Agenda
Tab B List of Participants

SYSTEM II
90939
Add-on

12

SECRET

NATIONAL SECURITY COUNCIL MEETING

Friday, September 20, 1985
11:00 a.m. - 12:00 Noon
Cabinet Room

AGENDA

I. Introduction.....Robert C. McFarlane
(5 minutes)

II. Objectives for Geneva Meetings & perspectives of the Shevardnadze Meeting ..... Secretary Shultz
(10 minutes)

III. Intelligence review on Soviet expectations..... William P. Casey
(5 minutes)

IV. Defense perspective..... Secretary Weinberger
(5 minutes)

V. Discussion.....All Participants
(30 minutes)

VI. Conclusion.....Robert C. McFarlane
(5 minutes)

~~SECRET~~
Declassify on: OADR

# LIST OF PARTICIPANTS

The President

The Vice President
Mr. Craig L. Fuller

State
Secretary George P. Shultz
Ms. Rozanne L. Ridgway (Assistant Secretary for European Affairs)

Treasury
Secretary James A. Baker, III
Mr. Robert M. Kimmitt

Justice
Attorney General Edwin Meese III

CIA
Mr. William J. Casey
Mr. Robert M. Gates

OMB
Dr. Alton Keel

JCS
General John A. Wickham Jr. (Acting Chairman)
Admiral Arthur S. Moreau

White House
Mr. Donald T. Regan
Mr, Robert C. McFarlane
Admiral John M. Poindexter

NSC
Amb Jack F. Matlock
Colonel Tyrus Cobb

7816 JFC 14

SYSTEM II
90939
Add-on #2

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

~~SECRET~~

September 19, 1985

ACTION

MEMORANDUM FOR ROBERT C. MCFARLANE

FROM: TYRUS W. COBB/JACK F. MATLOCK

SUBJECT: Your Talking Points for Shevardnadze NSC Meeting, Friday, September 20, 1985 -- 11:00 a.m.

Attached at Tab A are suggested talking points for your use for the Shevardnadze NSC meeting tomorrow. As we had discussed you would begin this session with an introduction highlighting to the President that this session is intended to serve as a strategic overview of our broad policy toward the Soviet Union and to examine the major issues on your bilateral, regional and human rights agenda. In addition, we recommend that you review for the President our public diplomacy strategy and highlight the key events between now and November 18 that present opportunities for us to convey our position convincingly to the American and Allied publics -- and to the Soviet leadership. You might then turn to Secretary Shultz who will follow with a review of the key Soviet-American issues in the three areas and delineate our objectives for the meeting with Gorbachev. Bill Casey is prepared to follow with a five minute presentation on Soviet objectives, followed by Cap who will treat Soviet defense concerns generally.

The Agenda for the NSC meeting is at Tab B.

Ron Lehman, Bob Linhard and Johnathan Miller concur.

RECOMMENDATION

That you review the suggested talking points at Tab A.

Approve _____ Disapprove _____

Attachments:

Tab A Suggested Talking Points

Tab B Proposed Agenda

~~SECRET~~
Declassify on: OADR

7817

15

SECRET

TALKING POINTS FOR SHEVARDNADZE NSC MEETING

-- Mr. President, this NSC meeting should serve as a forum to provide you with a strategic overview of the broad direction we hope to pursue in dealing with the Soviet leadership over the next two months. We will also review the major issues on our bilateral, human rights, and regional agendas. We would prefer to hold off any analysis of arms control issues until next week.

-- Before I turn to George I would like to briefly review our public diplomacy approach to the Geneva meeting and the key events around which our strategy will be focused.

-- Five public diplomacy milestones will generate the most media and public interest. We will reinforce with briefings, backgrounders and public statements.

1. George's September 24 speech to the UNGA.
2. His meeting the following day with Soviet Foreign Minister Shevardnadze and the Soviet FM's own speech to the UN.
3. Your meeting here with Shevardnadze on September 27.
4. Gorbachev's October 2-5 visit to France.
5. Your speech to the UNGA and that of Soviet President Andrei Gromyko if he comes to New York.



SECRET

16

-- Believe that Gorbachev will trigger a new round of media interest when he visits France. Judgment is that his novelty value is wearing off. He has failed to convince European publics that his many proposals are more than old Soviet propaganda in new packaging. You will be giving a written interview to a French newspaper, Le Figaro, to get your own views before that same audience.

-- We will stress that our Agenda for the meeting includes arms control, but goes much farther in an attempt to engage the Soviets on the many other sources of tension between us, including Human Rights, Afghanistan and their expansionist policies. We want, in other words, to discuss the sources of tension between us as well as the symptoms of it.

-- We will want to demonstrate the Soviets' responsibility for Geneva results. Will stress that we want to make the meeting a success but we have no illusions. We are neither naive nor negative in our approach.

-- Finally, Mr. President, your own role will include, in addition to your UNGA speech and your November press conference -- a series of press lunches with influential newsmen, a number of television and radio interviews for both foreign and domestic media, and a speech to the world via Worldnet. We hope to organize a well-publicized meeting with congressional leaders shortly before your departure.

17

-- Mr. President, George Shultz will now provide a review of the major Soviet-American issues in the three areas and delineate our key objectives for the meeting with Shevardnadze and, later, with Gorbachev.

-- Bill Casey will now provide his perspective on Soviet expectations and objectives for these meetings.

-- Cap will now present another perspective, focusing on key security concerns.

-- I would like now to open this session to general discussion, Mr. President,to focus not only on the key issues in the areas we have discussed, but to stimulate our thinking with respect to our overall approach toward the Soviet Union.

-- Mr. President, this concludes your briefing.

7818

SYSTEM II
90939
Add-on #2

18

SECRET

NATIONAL SECURITY COUNCIL MEETING

Friday, September 20, 1985
11:00 a.m. - 12:00 Noon
Cabinet Room

AGENDA

I. Introduction.....Robert C. McFarlane
(5 minutes)

II. Objectives for Geneva Meetings and Perspectives of the Shevardnadze Meeting.....Secretary Shultz
(10 minutes)

III. Intelligence Review on Soviet Expectations.....William P. Casey
(5 minutes)

IV. Defense Perspective.....Secretary Weinberger
(5 minutes)

V. Discussion.....All Participants
(30 minutes)

VI. Conclusion.....Robert C. McFarlane
(5 minutes)

SECRET
Declassify on: OADR

19

SYSTEM II
90939
Add-on #2

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

~~SECRET~~

September 19, 1985

ACTION

MEMORANDUM FOR ROBERT C. MCFARLANE

FROM: TYRUS W. COBB/JACK F. MATLOCK

SUBJECT: Your Talking Points for Shevardnadze NSC Meeting, Friday, September 20, 1985 -- 11:00 a.m.

Attached at Tab A are suggested talking points for your use for the Shevardnadze NSC meeting tomorrow. As we had discussed you would begin this session with an introduction highlighting to the President that this session is intended to serve as a strategic overview of our broad policy toward the Soviet Union and to examine the major issues on your bilateral, regional and human rights agenda. In addition, we recommend that you review for the President our public diplomacy strategy and highlight the key events between now and November 18 that present opportunities for us to convey our position convincingly to the American and Allied publics -- and to the Soviet leadership. You might then turn to Secretary Shultz who will follow with a review of the key Soviet-American issues in the three areas and delineate our objectives for the meeting with Gorbachev. Bill Casey is prepared to follow with a five minute presentation on Soviet objectives, followed by Cap who will treat Soviet defense concerns generally.

The Agenda for the NSC meeting is at Tab B.

Ron Lehman, Bob Linhard and Johnathan Miller concur.

RECOMMENDATION

That you review the suggested talking points at Tab A.

Approve _____ Disapprove _____

Attachments:

Tab A Suggested Talking Points

Tab B Proposed Agenda

~~SECRET~~
Declassify on: OADR

7820 28

SECRET

TALKING POINTS FOR SHEVARDNADZE NSC MEETING

-- Mr. President, this NSC meeting should serve as a forum to provide you with a strategic overview of the broad direction we hope to pursue in dealing with the Soviet leadership over the next two months. We will also review the major issues on our bilateral, human rights, and regional agendas. We would prefer to hold off any analysis of arms control issues until next week.

-- Before I turn to George I would like to briefly review our public diplomacy approach to the Geneva meeting and the key events around which our strategy will be focused.

-- Five public diplomacy milestones will generate the most media and public interest. We will reinforce with briefings, backgrounders and public statements.

1. George's September 24 speech to the UNGA.
2. His meeting the following day with Soviet Foreign Minister Shevardnadze and the Soviet FM's own speech to the UN.
3. Your meeting here with Shevardnadze on September 27.
4. Gorbachev's October 2-5 visit to France.
5. Your speech to the UNGA and that of Soviet President Andrei Gromyko if he comes to New York.

21

-- Believe that Gorbachev will trigger a new round of media interest when he visits France. Judgment is that his novelty value is wearing off. He has failed to convince European publics that his many proposals are more than old Soviet propaganda in new packaging. You will be giving a written interview to a French newspaper, Le Figaro, to get your own views before that same audience.

-- We will stress that our Agenda for the meeting includes arms control, but goes much farther in an attempt to engage the Soviets on the many other sources of tension between us, including Human Rights, Afghanistan and their expansionist policies. We want, in other words, to discuss the sources of tension between us as well as the symptoms of it.

-- We will want to demonstrate the Soviets' responsibility for Geneva results. Will stress that we want to make the meeting a success but we have no illusions. We are neither naive nor negative in our approach.

-- Finally, Mr. President, your own role will include, in addition to your UNGA speech and your November press conference -- a series of press lunches with influential newsmen, a number of television and radio interviews for both foreign and domestic media, and a speech to the world via Worldnet. We hope to organize a well-publicized meeting with congressional leaders shortly before your departure.

-- Mr. President, George Shultz will now provide a review of the major Soviet-American issues in the three areas and delineate our key objectives for the meeting with Shevardnadze and, later, with Gorbachev.

-- Bill Casey will now provide his perspective on Soviet expectations and objectives for these meetings.

-- Cap will now present another perspective, focusing on key security concerns.

-- I would like now to open this session to general discussion, Mr. President,to focus not only on the key issues in the areas we have discussed, but to stimulate our thinking with respect to our overall approach toward the Soviet Union.

-- Mr. President, this concludes your briefing.

7821

23

SYSTEM II
90939
Add-on #2

SECRET

NATIONAL SECURITY COUNCIL MEETING

Friday, September 20, 1985
11:00 a.m. - 12:00 Noon
Cabinet Room

AGENDA

I. Introduction.....Robert C. McFarlane
(5 minutes)

II. Objectives for Geneva Meetings and Perspectives of the Shevardnadze Meeting.....Secretary Shultz
(10 minutes)

III. Intelligence Review on Soviet Expectations.....William P. Casey
(5 minutes)

IV. Defense Perspective.....Secretary Weinberger
(5 minutes)

V. Discussion.....All Participants
(30 minutes)

VI. Conclusion.....Robert C. McFarlane
(5 minutes)

~~SECRET~~
Declassify on: OADR

7402

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

September 19, 1985

ACTION

MEMORANDUM FOR ROBERT C. MCFARLANE

THROUGH: WILLIAM F. MARTIN

FROM: JACK F. MATLOCK

SUBJECT: Request to Travel to New York to Participate in the Meeting with Secretary Shultz and Soviet Foreign Minister Shevardnadze, September 25, 1985

I plan to travel to New York to participate in the meeting with Secretary Shultz and Soviet Foreign Minister Eduard Shevardnadze to be held on September 25, 1985. Transportation and per diem costs to be paid by NSC.

RECOMMENDATION

That you approve my travel.

Approve _____ Disapprove _____

cc: Administrative Office

THE WHITE HOUSE
WASHINGTON

NOT FOR System
T Johnson

25

Eyes Only
F Mallork

NON-LOG
~~OUTSIDE THE SYSTEM~~ Chron

26

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

~~CONFIDENTIAL~~/EYES ONLY

September 20, 1985

INFORMATION

MEMORANDUM FOR JOHN M.POINDEXTER

FROM: JACK F. MATLOCK

SUBJECT: Your Meeting with Tom Johnson, Saturday, September 21, 11:00 a.m.

You have agreed to meet with LTC Tom Johnson from West Point and me to discuss technical aspects of the SDI program. Tom is the director of the Science Research Lab at the Academy and has more than 20 years experience in R¢D and strategic defense matters. Johnson holds a PhD from LLL/University of California and is a protege of Edward Teller, Jonny Foster, both of whom he continues to work closely with on SDI issues. He served as a Special Assistant to Jay Keyworth in 1981-82 (but the relationship soured over matters relating to SDI projects) and serves as an advisor to the SDIP in many capacities. For that reasonn I would appreciate it if you would hold this meeting very close.

By the way, Tom is also a protege of Archibald McLeish and is a well-published poet. You may also be interested to know that he recently returned from a visit to the USSR where he toured many of their labs at the invitation of Velikhov.

I would suggest that we ask Tom to provide you with a candid assessment of the technical viability of possible SDI systems, identify what are the most promising SDI technologies, and discuss implications for our negotiating position.

I know you have a particular interest in the software aspects of the SDI program. Attached is the letter of resignation from Professor Parnas who expressed serious concern with the technical viability of the computing requirements for a successful SDI program and with aspects of the management of the effort. In addition a few of his brief papers are attached that address facets of the problem. Tom will also be prepared to discuss these if you wish.

Attachment:

TAB A Letter of Resignation with Enclosures

~~CONFIDENTIAL~~/EYES ONLY
Declassify on: OADR

27

OUTSIDE THE SYSTEM

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

~~CONFIDENTIAL/EYES ONLY~~ September 20, 1985

INFORMATION

MEMORANDUM FOR JOHN M.POINDEXTER

FROM: JACK F. MATLOCK

SUBJECT: Your Meeting with Tom Johnson, Saturday, September 21, 11:00 a.m.

You have agreed to meet with LTC Tom Johnson from West Point and me to discuss technical aspects of the SDI program. Tom is the director of the Science Research Lab at the Academy and has more than 20 years experience in R¢D and strategic defense matters. Johnson holds a PhD from LLL/University of California and is a protege of Edward Teller, Jonny Foster, both of whom he continues to work closely with on SDI issues. He served as a Special Assistant to Jay Keyworth in 1981-82 (but the relationship soured over matters relating to SDI projects) and serves as an advisor to the SDIP in many capacities. For that reasonn I would appreciate it if you would hold this meeting very close.

By the way, Tom is also a protege of Archibald McLeish and is a well-published poet. You may also be interested to know that he recently returned from a visit to the USSR where he toured many of their labs at the invitation of Velikhov.

I would suggest that we ask Tom to provide you with a candid assessment of the technical viability of possible SDI systems, identify what are the most promising SDI technologies, and discuss implications for our negotiating position.

I know you have a particular interest in the software aspects of the SDI program. Attached is the letter of resignation from Professor Parnas who expressed serious concern with the technical viability of the computing requirements for a successful SDI program and with aspects of the management of the effort. In addition a few of his brief papers are attached that address facets of the problem. Tom will also be prepared to discuss these if you wish.

Attachment:

TAB A Letter of Resignation with Enclosures

~~CONFIDENTIAL/EYES ONLY~~
Declassify on: OADR

28

UNIVERSITY OF VICTORIA

P.O. BOX 1700, VICTORIA, BRITISH COLUMBIA, CANADA V8W 2Y2
TELEPHONE (604) 721-7211, TELEX 049-7222

*Department of Computer Science*
*721-7209*

2 July, 1985

Dr. George Keyworth
Science Advisor to the President
Office of Science and Technology Policy
Executive Office of the President
Washington, D.C.
20506
USA

Dear Dr. Keyworth:

I recently resigned my membership in the SDIO Panel on Computing in Support of Battle Management. I am sending you a copy of my letter of resignation together with 8 very short papers that were enclosed to support my position with more technical arguments. As I state in the letter, I believe that the President should be fully informed on this matter.

I have tried to remove all technical jargon from these papers and to make their content fundamental enough that scientists who are not experts in computing can follow them. If I can be of help to you in understanding them, please let me know.

Yours truly,

David L. Parnas

David L. Parnas
Lansdowne Professor

DLP:bls

29


# UNIVERSITY OF VICTORIA

P.O. BOX 1700, VICTORIA, BRITISH COLUMBIA, CANADA V8W 2Y2
TELEPHONE (604) 721-7211, TELEX 049-7222

*Department of Computer Science*
*721-7209*


28 June, 1985

Mr. James H. Offut
Assistant Director, BM/C3
Strategic Defense Initiative Organization
Office of the Secretary of Defense
Washington, D.C.
20301

Dear Mr. Offut:

Thank you for your letter of 5 June 1985 appointing me a member of the SDIO Panel on Computing in Support of Battle Management. I appreciate the recognition implicit in being chosen as one of your expert advisors on computer science.

After attending the first meeting of the panel and giving the problem considerable thought, I am resigning my membership in the panel. I do not believe that further work by the panel will be useful and I cannot, in good conscience, accept further payment for useless effort.

The panel's work will not be useful for two reasons.

1) The goals stated for the Strategic Defense System cannot be attained by the class of systems that you are considering.

2) The SDIO is not the appropriate organization to fund and administer the research it is supporting. Most of the money spent will be wasted. The panel on which you have asked me to serve, is not appropriately constituted, clearly chartered, and adequately informed. There are better ways to select and manage research.

My conclusions are not based on political or policy judgements. Unlike many other academic critics of the SDI effort, I have not, in the past, objected to defense efforts or defense sponsored research. I have been deeply involved in such research and have consulted extensively on defense projects. My conclusions are based on more than 20 years of research on software engineering including more than 8 years of work on real-time software used in military aircraft. They are based on familiarity with both operational military software and computer science research. My conclusions are based on characteristics peculiar to this particular effort, not objections to weapons development in general.

Before making my decision and writing this letter I have carefully reconsidered what I have learned in my own research area and I have reviewed reports of work in related fields. These reviews lead inevitably to the judgements stated above. I am willing to stake my professional reputation on my conclusions.

Enclosed with this letter are several brief papers (1 - 2 pages each) summarizing my observations and substantiating the conclusions stated above. Their purpose is to explain my decision.

These papers explain:

1) The fundamental technological differences between software engineering and other areas of engineering and why software is unreliable,

2) The properties of the proposed SDI software that make it unattainable,

3) Why the techniques commonly used to build military software are inadequate for this job,

4) The nature of research in Software Engineering, and why the improvements that it can effect will not be sufficient to allow construction of a truly reliable strategic defense system,

5) The nature of research in Artificial Intelligence, and why I do not expect it to help in building reliable military software,

6) The history of research in Automatic Programming, and why I do not expect it to bring about the substantial improvements that are needed,

7) Why Program Verification cannot give us a reliable strategic defense battle management software system,

8) My opinions on the management of applied research, why I consider this panel and the SDIO in general to be an inappropriate vehicle for funding research, and what I would do instead.

I am quite certain that you will be able to find software experts who disagree with my conclusions. For many, the project offers a source of funding, funding that will enrich some personally, while offering others new and generous support for their personal research projects. During the first sittings of our panel, I could see the dollar figures dazzling everyone involved. Almost everyone that I know within the military industrial complex sees in the SDI a new "pot of gold" just waiting to be tapped.

For others, the project offers an unending set of technological puzzles that are fun to work on; such problems are exciting and challenging whether or not the work ever produces useful results. Almost every software expert that I know, entered the field because they enjoy this kind of challenge. Several of the speakers at the first meeting of our panel could not hide their delight at the unbounded set of technical challenges implicit in the unattainable goals of the project.

I can tell you, as one who likes both money and technical challenges, that these temptations are very hard to resist. You will find it very hard to find unbiased expert opinions on this issue.

In March 1983 the President asked us, as members of the scientific community, to provide the means of rendering nuclear weapons impotent and obsolete. I believe that it is our duty, as scientists and engineers, to reply that we have no technological magic that will accomplish that. The short term applied research and focussed development that SDI is now funding is not going to solve the problem; the President and the public should know that.

Yours truly,

David L. Parnas
Lansdowne Professor

DLP:jcs

Cc: S. Wilson, panel members

# WHY SOFTWARE IS UNRELIABLE

David Lorge Parnas
University of Victoria*

## I. Introduction

People familiar with both software engineering and older engineering disciplines observe that the state of the art in software is significantly behind that in other areas of engineering. When most engineering products have been completed, tested and sold, it is reasonable to expect that the product design is correct and that it will work reliably. With software products, it is usual to find that the software has major "bugs" and does not work reliably for some users. These problems may persist for several versions and sometimes worsen as the software is "improved". While most products come with an express or implied warranty, software products often carry a specific disclaimer of warranty. The lay public, familiar with only a few incidents of software failure, may regard them as exceptions caused by exceptionally inept programmers. Those of us who are software professionals, know better; the most competent programmers in the world cannot avoid such problems. This note discusses one technical reason for this situation.

## II. System Types

Engineering products can be classified as either discrete state systems, analog systems, or hybrid systems.

Discrete state or digital systems are made from components with a finite number of stable states. They are designed in such a way that the behavior of the system when not in a stable state is not significant.

Continuous or analog systems are built of components that, within a broad operating range, have an infinite number of stable states and whose behavior can be adequately described by continuous functions.

Hybrid systems are mixtures of the two types of components. For example, we may have an electrical circuit containing, in addition to analog components, a few components whose descriptive equations have discontinuities (e.g. diodes). Each of these components has a small number of discrete operating states. Within those states its behavior can be described by continuous functions.

## III. Mathematical tools

Analog systems form the core of the traditional areas of engineering. The mathematics of continuous functions is well understood. When we say that a system is described by continuous functions we are saying that it can contain no hidden surprises. Small changes in inputs will always cause correspondingly small changes in outputs. An engineer who insures, through careful design, that the system components are always operating within their normal operating range, can use a mathematical analysis to insure that there are no surprises. When combined with testing to insure that the components are within their operating range, this leads to reliable systems.

Before the advent of digital computers, when discrete state systems were built, the number of states in such systems was relatively small. With a small number of states, exhaustive analysis and exhaustive testing was possible. Such testing compensated for the lack of mathematical tools

*Also with Naval Research Laboratory, Washington, D.C.

corresponding to those used in analog systems design. The engineers of such systems still had systematic methods that allowed them to obtain a complete understanding of their system's behavior.

The design of many hybrid systems can be verified by a combination of the two methods. We can then identify a finite number of operating states for the components with discrete behavior. Within those states, the system's behavior can be described by continuous functions. Usually the number of states that must be distinguished is small. For each of those states, the tools of continuous mathematics can be applied to analyze the behavior of the system.

With the advent of digital computers, we found the first discrete state systems with very large numbers of states. However, to manufacture such systems it was necessary to construct them using many copies of very small digital subsystems. Each of those small subsystems could be analyzed and tested exhaustively. Because of the repetitive structure, exhaustive testing was not necessary to obtain correct and reliable hardware. While design errors are found in computer hardware, they are considered exceptional. They usually occur in those parts of the computer that are not repetitive structures.

Software systems are discrete state systems that do not have the repetitive structure found in computer circuitry. There is seldom a good reason to construct software as highly repetitive structures. The number of states in software systems is orders of magnitude larger than the number of states in the non-repetitive parts of computers. The mathematical functions that describe the behavior of these systems are not continuous functions and traditional engineering mathematics does not help in their verification. This difference clearly contributes to the relative unreliability of software systems and the apparent lack of competence of software engineers. It is a fundamental difference that will not disappear with improved technology.

## IV. How can we understand software?

To ameliorate the problems caused by this fundamental difference in technology two techniques are available, (a) the building of software as highly organized collections of small programs, (b) the use of mathematical logic to replace continuous mathematics.

Dividing software into modules and building each module of so called "structured" programs, clearly helps. When properly done, each component deals with a small number of cases and can be completely analyzed. However, real software systems have many such components and there is no repetitive structure to simplify the analysis. Even in highly structured systems, surprises and unreliability occur because the human mind is not able to fully comprehend the many conditions that can arise because of the interaction of these components. Moreover, finding the right structure has proven to be very difficult. Well structured real software systems are still very rare.

Logic is a branch of mathematics that can deal with functions that are not continuous. Many researchers believe that it can play the role in software engineering that continuous mathematics plays in mechanical and electrical engineering. Unfortunately, this has not yet been verified in practice. The large number of states and lack of regularity in the software results in extremely complex mathematical expressions. Disciplined use of these expressions is beyond the computational capacity of both the human programmer and current computer systems. There is progress in this area but it is very slow and we are far from being able to handle even small software systems. With current techniques the mathematical expressions describing a program are often notably harder to understand than the program itself.

## V. The education of programmers

Worsening the differences between software and other areas of technology is a personnel problem. Most designers in traditional engineering disciplines have been educated to understand the mathematical tools that are available to them. Most programmers cannot even begin to use the meager tools that are available to software engineers.

33

# WHY THE SDI SOFTWARE SYSTEM WILL BE UNTRUSTWORTHY

David Lorge Parnas
University of Victoria*

## I. Introduction

In March 1983, the President called for an intensive and comprehensive effort to define a long-term research program with the ultimate goal of eliminating the threat posed by nuclear ballistic missiles. He asks us, as members of the scientific community, to provide the means of rendering these nuclear weapons impotent and obsolete. To accomplish this goal we would need a software system so well developed that we could have extremely high confidence that the system would work correctly when called upon. In the sequel I will present some of the characteristics of the required battle management software and then discuss their implications on the feasibility of achieving that confidence.

## II. Characteristics of the proposed Battle Management Software System

1) The system will be required to identify, track, and direct weapons towards targets whose ballistic characteristics cannot be known with certainty before the moment of battle. It must distinguish these targets from decoys whose characteristics are also unknown.

2) The computing will be done by a network of computers connected to sensors, weapons, and each other, by channels whose behavior, at the time the system is invoked, cannot be predicted because of possible countermeasures by an attacker. The actual subset of system components that will be available at the time that the system is put into service, and throughout the period of service, cannot be predicted for the same reason.

3) It will be impossible to test the system under realistic conditions prior to its actual use.

4) The service period of the system will be so short that there will be little possibility of human intervention and no possibility of debugging and modification of the program during that period of service.

5) Like many other military programs, there are absolute real-time deadlines for the computation. The computation will consist primarily of periodic processes but the number of those processes that will be required, and the computational requirements of each process, cannot be predicted in advance because they depend on target characteristics. The resources available for computation cannot be predicted in advance. We cannot even predict the "worst case" with any confidence.

6) The weapon system will include a large variety of sensors and weapons, most of which will themselves require a large and complex software system. The suite of weapons and sensors is likely to grow during development and after deployment. The characteristics of weapons and sensors are not yet known and are likely to remain fluid for many years after deployment. The result is that the overall battle management software system will have to integrate a software system significantly larger than has ever been attempted before. The components of that system will be subject to independent modification.

---

*Also with Naval Research Laboratory, Washington, D.C.



26 June, 1985

III. Implications of these problem characteristics

Each of these characteristics has clear implications on the feasibility of building battle management software that will meet the President's requirements.

1) Fire control software cannot be written without making assumptions about the characteristics of enemy weapons and targets. This information is used in determining the recognition algorithms, the sampling periods, and the noise-filtering techniques. If the system is developed without knowledge of these characteristics, or with the knowledge that the enemy can change some of them on the day of battle, there are likely to be subtle but fatal errors in the software.

2) Although there has been some real progress in the area of "fail-soft" computer software, I have seen no success except in situations where (a) the likely failures can be predicted on the basis of past history, (b) the component failures are unlikely and are statistically independent, (c) the system has excess capacity, (d) the real-time deadlines, if any, are soft, i.e. they can be missed without long term effects. None of these are true for the required battle management software.

3) No large scale software system has ever been installed without extensive testing under realistic conditions. For example, in operational software for military aircraft, even minor modifications require extensive ground testing followed by flight testing in which battle conditions can be closely approximated. Even with these tests, bugs can and do show up in battle conditions. The inability to test a strategic defense system under field conditions before we actually need it, will mean that no knowledgeable person would have much faith in the system.

4) It is not unusual for software modifications to be made in the field. Programmers are transported by helicopter to Navy ships; debugging notes can be found on the walls of trucks carrying computers that were used in Vietnam. It is only through such modifications that software becomes reliable. Such opportunities will not be available in the 30 minute war to be fought by a strategic defense battle management system.

5) Programs of this type must meet hard real-time deadlines reliably. In theory, this can be done either by scheduling at runtime or by pre-runtime scheduling. In practice, efficiency and predictability require some pre-runtime scheduling. Schedules for the worst case load are often built into the program. Unless one can work out worst case real-time schedules in advance, one can have no confidence that the system will meet its deadlines when its service is required.

6) All of our experience indicates that the difficulties in building software increase with the size of the system, with the number of independently modifiable subsystems, and the number of interfaces that must be defined. Problems worsen when the interfaces may change. The consequent modifications increase the complexity of the software and the difficulty of making a change correctly.

IV. Conclusion

All of the cost estimates indicate that this will be the most massive software project ever attempted. The system has numerous technical characteristics that will make it more difficult than previous systems, independent of size. Because of the extreme demands on the system and our inability to test it, we will never be able to believe, with any confidence, that we have succeeded. Nuclear weapons will remain a potent threat.

# WHY CONVENTIONAL SOFTWARE DEVELOPMENT DOES NOT PRODUCE RELIABLE PROGRAMS

**David Lorge Parnas**
University of Victoria*

## I. What is the conventional method?

The easiest way to describe the programming method used in most projects today was given to me by a teacher who was explaining how he teaches programming. "Think like a Computer", he said. He instructed his students to begin by thinking about what the computer had to do first and to write that down. They would then think about what the computer had to do next and continue in that way until they had described the last thing that the computer would do. This, in fact, is the way that I was taught to program. Most of today's textbooks demonstrate the same method, although it has been improved by allowing us to describe the computer's "thoughts" in larger steps and later to refine those large steps to a sequence of smaller steps.

## II. Why this method leads to confusion

This intuitively appealing method works well - on problems too small to matter. We think that it works because it worked for the first program that we wrote. One can follow the method with programs that have neither branches nor loops. As soon as our thinking reaches a point where the action of the computer must depend on conditions that are not known until the program is running, we must deviate from the method by labeling one or more of the actions and remembering how we would get there. As soon as we introduce loops into the program there are several ways of getting to some of the points and we must remember all of those ways. As we progress through the algorithm, we recognize the need for information about earlier events and add variables to our data structure. We now have to start remembering what the data means and under what circumstances it is meaningful. As we continue in our attempt to "think like a computer", the amount that we have to remember grows and grows. The simple rules defining how we got to certain points in a program become more complex as we branch there from other points. The simple rules defining what the data means become more complex as we find other uses for existing variables and add new variables. Eventually, we make an error. Sometimes we note that error, sometimes it is not found until we test. Sometimes the error is not very important; it only happens on rare or unforeseen occasions. In that case, we find it when the program is in use. Often, because one needs to remember so much about the meaning of each label and each variable, new problems are created when old problems are corrected.

## III. What is the effect of concurrency on this method?

In many of our computer systems there are several sources of information and several outputs that must be controlled. This leads to a computer that might be thought of as doing many things at once. If the sequence of external events cannot be predicted in advance, the sequence of actions taken by the computer is also not predictable. The computer may only be doing one thing at a time but as one attempts to "think like a computer" one finds many more points where the action must be conditional on what happened in the past. Any attempt to design these programs by thinking things through in the order that the computer will execute them leads to confusion and results in systems that nobody can understand completely.

---

*Also with Naval Research Laboratory, Washington, D.C.

IV. What is the effect of multi-processing

When there is more than one computer in a system, the software not only appears to be doing more than one thing at a time, it really is doing many things at once. There is no sequential program that one can study. Any attempt to "think like the computer system" is obviously hopeless. There are so many possibilities to consider, that only extensive testing can begin to sort things out. Even after such testing we have incidents such as occurred on a Space Shuttle flight several years ago. The wrong combination of sequences occurred and prevented the flight from starting.

V. Do Professional Programmers really use this approach?

Yes.

I have had occasion to study lots of practical software and to discuss programs with lots of professional programmers. In recent years many programmers have tried to improve their working methods using a variety software design approaches. However, when they get down to writing executable programs, they revert to the conventional way of thinking. I have yet to find a substantial program in practical use whose structure was not based on the expected execution sequence. I would be happy to be shown some.

Other methods are discussed in advanced courses, a few good textbooks, and scientific meetings, but the majority of our programmers continue to use the basic approach of thinking things out in the order that the computer will execute them. This is most noticeable in the maintenance (deficiency correction) phase of programming.

VI. How do we get away with this inadequate approach?

It should be clear that writing and understanding very large real-time programs by "thinking like a computer" will be beyond our intellectual capabilities. How can it be that we have so much software that is reliable enough for us to use it? The answer is simple; programming is a trial and error craft. People write programs without any expectation that they will be right the first time. They spend at least as much time testing them and correcting errors as they spent writing the initial program. Large concerns have separate groups of testers to do quality assurance. Programmers cannot be trusted to test their own programs adequately. Software is released for use, not when it is known to be correct, but when the rate of discovering new errors slows down to one that management considers acceptable. Users learn to expect errors and are often told how to avoid the bugs until the program is improved.

VII. Conclusion

The military software that we depend on every day is not likely to be correct. The methods that are in use in the industry today are not adequate for building large real-time software systems that must be reliable when first used. A drastic change in methods is needed.

# THE LIMITS OF SOFTWARE ENGINEERING METHODS

**David Lorge Parnas**
**University of Victoria***

## I. What is Software Engineering Research

We have known for 25 years that our programming methods are inadequate for large projects. Research in Software Engineering, Programming Methodology, Software Design, etc., looks for better tools and methods. The common thrust of results in these fields is to reduce the amount that a programmer must remember when checking and changing a program.

Two main lines of research are, (1) structured programming and the use of formal program semantics, (2) the use of formally specified abstract interfaces to hide information about one module (work assignment) from the programmers who are working on other parts. A third idea, less well understood but no less important, was the use of cooperating sequential processes to help deal with the complexities arising from concurrency and multi-programming. By the late 1970's the basic ideas in software engineering were considered "motherhood" in the academic community. Nonetheless, examinations of real programs revealed that actual programming practice, especially for real-time systems, had not been changed much by the publication of the academic proposals.

The gap between theory and practice was large and growing. Those espousing structured approaches to software were certain that it would be easy to apply their ideas in real systems. Programmers working on "real" software did not see how to apply these ideas to the problems that they faced in their daily work. They doubted that programs organized according to the principles espoused by the academics could ever meet the performance constraints on "real" systems. Even those who claimed to believe in those principles were not able to apply them consistently.

In 1977 the management Naval Research Laboratory in Washington and the Naval Weapons Center in China Lake California, decided that something should be done to close the gap. They asked one of the academics who had faith in the new approaches (myself) to demonstrate the applicability of those methods by building, for comparisons sake, a second version of a Navy real-time program. The project, now known as the Software Cost Reduction project (SCR), was expected to take 2 - 4 years. It is still going on.

The project has made two things clear, (1) much of what the academics proposed can be done, (2) good software engineering is far from easy. The methods reduce, but do not eliminate, errors. They reduce, but do not eliminate, the need for testing.

## II. What should we do and what can we do?

The SCR work has been based upon the following precepts:

1) Before starting software design, nail the software requirements down with a complete, black-box, requirements document,

2) Before starting to write programs divide the system into modules using information-hiding (abstraction),

3) Before writing the programs, each module should have a precise black-box formal specification,

---

*Also with Naval Research Laboratory, Washington, D.C.

4) Formal methods should be used to give precise documentation,

5) Real-time systems should be built as a set of cooperating sequential processes, each with a specified period and deadline,

6) Programs should be written using the ideas of structured programming as taught by Harlan Mills.

We have demonstrated that the first four of these precepts can be applied to military software by doing it. The documents that we have written have served as models for others. We have evidence that the models provide a most effective means of technology transfer.

We have not yet proven that these methods lead to reliable code that meets the space and time constraints. We have found that every one of these precepts is easier to pronounce than to carry out. Those who think that software design will become easy, and that errors will disappear, have not attacked substantial problems.

## III. What makes Software Engineering hard?

We can write software requirements documents that are complete and precise. We understand the mathematical model behind such documents and can follow a systematic procedure to document all necessary requirements decisions. Unfortunately, it is hard to make the decisions that must be made to write such a document. We often do not know how to make those decisions until we can play with the system. Only when we have built a similar system before, is it easy to determine the requirements in advance. It is worth doing, but it is not easy.

We know how to decompose complex systems into modules when we know the set of design decisions that must be made in the implementation. Each of these must be assigned to a single module. We can do that when we are building a system that resembles a system we built before. When we are solving a totally new problem, we will overlook difficult design decisions. The result will be a structure that does not fully separate concerns and minimize complexity.

We know how to specify abstract interfaces for modules. We have a set of standard notations for use in that task. Unfortunately, it is very hard to find the right interface. The interface should be an abstraction of the set of all alternative designs. We can find that abstraction only when we understand the alternative designs. For example, it has proven unexpectedly hard to design an abstract interface that hides the mathematical model of the earth's shape. We have no previous experience with such models and no one has designed such an abstraction before.

The common thread in all of these observations is that, even with sound software design principles, we need broad experience with similar systems to design good, reliable, software.

## IV. Will new programming languages make much difference?

Because of the very large improvements in productivity that were noted when compiler languages were introduced, many continue to look for another improvement by introducing better languages. Better notation always helps, but we cannot expect new languages to provide the same magnitude of improvement that we got from the first introduction of such languages. Our experience in SCR has not shown the lack of a language to be a major problem.

Programming languages are now sufficiently flexible that we can use almost any of them for almost any task. We should seek simplifications in programming languages, but we cannot expect that this will make a big difference.

V. What about programming environments?

The success of UNIX as a programming development tool has made it clear that the environment in which we work does make a difference. The flexibility of UNIX has allowed us to eliminate many of the time consuming housekeeping tasks involved in producing large programs. Consequently there is extensive research in programming environments. Here too, I expect small improvements can be made by basing tools on improved notations but no big breakthroughs. Problems with our programming environment have not been a major impediment in our SCR work.

VI. Why Software Engineering research will not make the SDI goals attainable?

Although I believe that further research on Software Engineering methods can lead to substantial improvements in our ability to build large real-time software systems, this work will not overcome the difficulties inherent in the plans for battle management computing for SDI. Software Engineering methods do not eliminate errors. They do not eliminate the basic differences between software technology and other areas of engineering. They do not eliminate either the need for extensive testing under field conditions or the need for opportunities to revise the system while it is in use. Most important, we have learned that the successful application of these methods depends on experience accumulated while building and maintaining similar systems. There is no such body of experience for SDI battle management.

VII. Conclusion

I am not a modest man. I believe that I have as sound and broad an understanding of the problems of Software Engineering as anyone that I know. If you gave me the job of building the system, and all the resources that I wanted, I could not do it. I don't expect the next 20 years of research to change that fact.

# ARTIFICIAL INTELLIGENCE AND THE STRATEGIC DEFENSE INITIATIVE

David Lorge Parnas
University of Victoria*

## I. Introduction

One of the technologies being considered for use in the SDI battle management software is Artificial Intelligence (AI). Researchers in AI have often made big claims and it is natural to believe that one should use this technology for a problem as difficult as SDI battle management. In this paper, I argue that one cannot expect much help from AI in building reliable battle management software.

## II. What is Artificial Intelligence?

Two quite different definitions of AI are in common use today.

AI-1: The use of computers to solve problems that previously could only be solved by applying human intelligence.

AI-2: The use of a specific set of programming techniques known as heuristic or rule based programming. In this approach human experts are studied to determine what heuristics or rules of thumb they use in solving problems. Usually they are asked for their rules. These rules are then encoded as input to a program that attempts to behave in accordance with them. In other words, the program is designed to solve a problem the way that humans seem to solve it.

It should be noted that the first definition defines AI as a set of problems, the second defines AI as a set of techniques.

The first definition has a sliding meaning. In the middle ages, it was thought that arithmetic required intelligence. Now we recognize it as a mechanical act. Something can fit the definition of AI-1 today but, once we see how the program works and understand the problem, we will not think of it as AI any more.

It is quite possible for a program to meet one definition and not the other. If we build a speech recognition program that uses Bayesian mathematics rather than heuristics it is AI-1 but not AI-2. If we write a rule based program to generate parsers for precedence grammers using heuristics it will be AI-2 but not AI-1 since the problem has a known algorithmic solution.

While it is possible for work to satisfy both definitions, the best AI-1 work that I have seen does not use heuristic or rule based methods. Workers in AI-1 often use traditional engineering and science approaches. They study the problem, its physical and logical constraints, and write a program that makes no attempt to mimic the way that people say they solve the problem.

## III. What can we learn from AI that will help us to build the battle management computer software?

I have seen some outstanding AI-1 work. Unfortunately, I cannot identify a body of techniques or technology that is unique to this field. When one studies these AI-1 programs one finds that they use sound scientific approaches, approaches that are also used in work that is not called AI. Most of the work is problem specific and some abstraction and creativity is required to see how to transfer it. People speak of AI as if it were some magic body of new ideas. There is good work in AI-1 but nothing so magic that it will allow the solution of the SDI battle management

*Also with Naval Research Laboratory, Washington, D.C.

problem.

I find the approaches taken in AI-2 to be dangerous and much of the work misleading. The rules that one obtains by studying people turn out to be inconsistent, incomplete, and inaccurate. Heuristic programs are developed by a trial and error process in which a new rule is added whenever one finds a case that is not handled by the old rules. This approach usually yields a program whose behavior is poorly understood and hard to predict. AI-2 researchers accept this evolutionary approach to programming as normal and proper. I trust such programs even less than I trust unstructured conventional programs. One never knows when the program will fail.

On occasion I have had to closely examine the claims of a worker in AI-2. I have always been disappointed. On close examination the heuristics turned out to handle a small number of obvious cases but failed to work in general. The author was able to demonstrate spectacular behavior on the cases that the program handled correctly. He marked the other cases as extensions for future researchers. In fact, the techniques being used often do not generalize and the improved program never appears.

## IV. What about Expert Systems?

Lately we hear a great deal about the success of a particular class of rule based systems known as expert systems. Every discussion of such systems cites one example of such a system that is being used to solve real problems by people other than its developer. That example is always the same - a program designed to find configurations for VAX computers. To many of us, that does not sound like a difficult problem; it sounds like the kind of problem that is amenable to algorithmic solution because VAX systems are constructed from well understood, well designed components. Recently I read a paper that reported that this program had become a maintenance nightmare. It was poorly understood, badly structured, and hence hard to change. I have good reason to believe that it could be replaced by a better program written using good software engineering techniques instead of heuristic techniques.

SDI presents a problem that may be more difficult than those being tackled in AI-1 and Expert Systems. Workers in those areas attack problems that now require human expertise. Some of the problems in SDI are problems where we now have no human experts. Do we now have humans who can, with high reliability and confidence, look at missiles in ballistic flight and distinguish warheads from decoys?

## V. Conclusion

Artificial Intelligence has the same relation to intelligence as Artificial Flowers have to Flowers. From a distance they may appear alike, but when closely examined they are quite different. I don't think that we can learn much about one by studying the other. AI offers no magic technology to solve our problems. Heuristic techniques do not yield systems that one can trust.

# CAN AUTOMATIC PROGRAMMING SOLVE THE SDI SOFTWARE PROBLEM

David Lorge Parnas
University of Victoria*

## I. Introduction

Throughout my career in computing I have read and heard people who claim that the solution to the software problem is automatic programming. All that one has to do is write the specifications for the software, and the computer will find a program. Can we expect such technology to produce reliable programs for SDI?

## II. Some perspective on Automatic Programming

The oldest paper that I know of that discusses Automatic Programming was written in 1948 by Saul Gorn when he was working at the Aberdeen Proving Ground. This paper, entitled "Is Automatic Programming Feasible", was classified for a while. It answered the question positively.

At that time, programs were fed into computers on paper tapes. The programmer worked the punch directly and actually looked at the holes in the tape. I have seen programmers "patch" programs by literally patching the paper tape.

The automatic programming system considered by Gorn in that paper was an assembler in today's terminology. All that one would have to do with his automatic programming system was to write a code such as CLA and the computer would automatically punch the proper holes in the tape. In this way, the programmer's task would be performed automatically by the computer.

In later years the phrase was used to refer to program generation from languages such as IT, FORTRAN, and ALGOL. In each case the programmer entered a specification of what he wanted, and the computer produced the program in the language of the machine.

In short, automatic programming always has been a euphemism for programming with a higher level language than was presently available to the programmer. Research in automatic programming is simply research in the implementation of higher level programming languages.

## III. Is automatic programming feasible? What does that mean?

Of course automatic programming is feasible. We have known for years that we can implement higher level programming languages. The only real question was the efficiency of the resulting programs. Usually, if the input "specification" is not a description of an algorithm, the resulting program is woefully inefficient. I do not believe that the use of non-algorithmic specifications as a programming language will prove practical for systems with limited computer capacity and hard real-time deadlines. When the input specification is a description of an algorithm, writing the specification is really writing a program. There will be no substantial change from our present capability.

## IV. Will automatic programming lead to more reliable programs?

The use of improved languages has led to a reduction in the amount of detail that a programmer must handle and hence to an improvement in reliability. However, extant programming languages, while far from perfect, are not that bad. Unless we move to non-algorithmic

*Also with Naval Research Laboratory, Washington, D.C.

specifications as an input to these systems, I do not expect a drastic improvement to result from this research.

On the other hand, our experience in writing non-algorithmic specifications has shown that people make mistakes in writing them just as they do in writing algorithms. The effect of such work on reliability is not yet clear.

V. Will automatic programming lead to a reliable SDI battle management system?

I believe that the claims that have been made for automatic programming systems are greatly exaggerated. Automatic programming in a way that is substantially different from what we do today is not likely to become a practical tool for real-time systems like the SDI battle management system. Moreover, one of the basic problems with SDI is that we do not have the information to write specifications that we can trust. In such a situation, automatic programming is no help at all.

# CAN PROGRAM VERIFICATION MAKE THE SDI SOFTWARE RELIABLE

David Lorge Parnas
University of Victoria*

## I. Introduction

Programs are mathematical objects. They have meanings that are mathematical objects. Program specifications are mathematical objects. Should it not be possible to prove that a program will meet its specification. This has been a topic of research now for at least 25 years. If we can prove programs correct, could we not prove the SDI software correct? If it was proven correct, could we not rely on it to defend us in time of need? These are the questions that we wish to answer in this little paper.

## II. What can we prove?

We can prove that certain small programs in special programming languages meet a specification. The word "small" is a relative one. Those working in verification would consider a 500 line program to be large. In discussing SDI software, we would consider a 500 line program to be small. The programs whose proofs I have seen have been well under 500 lines. They have performed easily defined mathematical tasks. They have been written without use of side effects, an important tool in practical programs.

Proofs for programs such as a model of the earth's gravity field do not have these properties. Such programs are larger, their specifications are not as neat or mathematically formalizable. They are often written in programming languages whose semantics are difficult to formalize. I have seen no proof of such a program.

Not only are manual proofs limited to programs of small size with mathematical specifications; machine theorem provers and verifiers are also strictly limited in the size of the program that they can handle. The size of programs that they can handle is several orders of magnitude different from the size of the programs that would constitute the SDI battle management system.

## III. Do we have the specifications?

In the case of SDI we do not have the specifications against which a proof could be applied. Even if size were not a problem, the lack of specifications would make the notion of a formal proof meaningless. If we wrote a formal specification for the software, we would have no way of proving that a program that satisfied that specification would actually do what we expected it to do. The specification might be wrong or incomplete.

## IV. Can we have faith in proofs?

Proofs increase our confidence in a program but we have no basis for complete confidence. Even in pure mathematics there are many cases of proofs that were published with errors. Proofs tend to be reliable when they are small, well polished and carefully read. They are not reliable when they are large, complex, and not read by anyone but their author. That is what would happen with any attempt to prove even a portion of the SDI software correct.

---

*Also with Naval Research Laboratory, Washington, D.C.



29 June, 1985

## V. What about concurrency?

The proof techniques that are most pra
work on proofs of systems of concurrent pr
rather than processes that cooperate using sh
shared memory. There are some techniques t
difficult than proofs for sequential programs or
ication over message channels.

## VI. What about programs that are supposed to

One of the major problems with the SDI
equipment destroyed or disabled by enemy actic
grams correct, I have seen only one attempt
answer in the event of a hardware failure. Th
We have no techniques for proving the corre
hardware failures and errors in input data.

## VII. Conclusion

It is inconceivable to me that one could pr
small portion of the SDI software. Given our ina
I do not know what such a proof would mean if I

40

# IS SDIO AN EFFICIENT WAY TO FUND WORTHWHILE RESEARCH

**David Lorge Parnas**
**University of Victoria***

## I. The proposal

In several discussions of this problem, I have found people telling me that they knew the SDIO software could not be built but felt that the project should continue because it might fund some good research. In this paper I want to discuss that point of view.

## II. The moral issue

There is an obvious moral problem with this position. The American people and their representatives have been willing to spend huge amounts of money on this project because of the hope that has been offered. Is it honest to take the attitude expressed above? Is it wise to have our policymakers make decisions on the assumption that such a system might be possible? I am not an expert on moral or political issues and offer no answer to these questions.

## III. Is DoD sponsoring of Software Research Effective?

I can raise another problem with this position. Is the SDIO an effective way to get good research done? Throughout many years of association with DoD I have been astounded at the amount of money that has been wasted on ineffective research projects. In my first contact with the U.S. Navy, I watched millions of dollars spent on a wild computer design that had absolutely no technical merit. It was abandoned many years after its lack of merit was clear. As a consultant for both the Navy and a number of contractors, I have seen expensive software research that produced very large reports with very little content. I have seen those large, expensive, reports put on shelves and never used. I have seen many almost alike efforts carried out independently and redundantly. I have seen talented professionals take approaches that they considered unwise because their "customers" asked for it. I have seen their customers take positions they do not understand because they thought that the contractors believed in them.

In computer software, the DoD contracting and funding scheme is remarkably ineffective because the bureaucrats who run it do not understand what they are buying.

## IV. Who can judge research?

The most difficult and crucial step in research is identifying and defining the problem. Successful researchers are usually those who have the insight to find a problem that is both solvable and important.

For applied research, additional judgement is needed. A problem may be an important one in theory, but there may be restrictions that prevent the use of its solution in practice. Only people closely familiar with the practical aspects of the problem can judge whether or not they could use the results of a research project.

Applied research must be judged by teams that include both successful researchers and experienced system engineers. They must have ample opportunity to meet, be fully informed, and have clearly defined responsibilities.

---

*Also with Naval Research Laboratory, Washington, D.C.

V. Who judges research in DoD?

Although there are a few notable exceptions within DoD, the majority of those who manage its applied research program are neither successful researchers nor people with extensive system building experience. There are outstanding researchers who work for DoD, but most of them work in the laboratories, not in the funding agencies. There are many accomplished system builders who work for DoD, but their managers often consider them too valuable to allow them to spend their time reviewing research proposals. The people who end up making funding decisions in DoD are very often unsuccessful researchers, unsuccessful system builders, and people who enter the bureaucracy immediately after their education. We call them technocrats.

Technocrats are bombarded with weighty volumes of highly detailed proposals that they are ill prepared to judge. They do not have time to study and think and are forced to rely on the advice of others. When they look for advice, they look for people that they know well, whether or not those are people whose areas of expertise are appropriate and whether or not those people can have unbiased positions on the subject.

Most technocrats are honest and hard-working, but they are not capable of doing what is needed.

The result is a very inefficient research program. I am convinced that there is now much more money being spent on software research than can usefully be spent. In spite of this, very little of the work that is sponsored leads to results that are useful. Of that, some of it goes unused because the good work is buried in the rest.

VI. The SDIO

The SDIO is a typical organization of technocrats. It is so involved in advocacy of the program that it cannot judge the quality of the research involved.

The SDIO panel on battle management computing contains not one person who has built actual battle management software. It contains no experts on trajectory computations, pattern recognition or other areas critical to this problem. All of its members stand to profit from continuation of the program.

VI. Alternatives

If there is good research being done by SDIO it has an applicability than is far broader than the SDI itself. That research should be managed by teams of scientists and engineers as part of a well organized research program. There is no need to create a special organization to judge this research. To do so is counterproductive. It can only make the program less efficient.

VII. Conclusion

There is no justification for continuing with the pretense that the SDI battle management software can be built just to obtain funding for otherwise worthwhile programs. DoD's overall approach to research management requires a thorough evaluation and review.

Classified

4Y

~~Sebra~~
Shcharansky

THE WHITE HOUSE
WASHINGTON

Matlock

Classified

**National Security Council**
**The White House**

49

*System #* _____

*Package #* _____

RETURN TO JACK MATLOCK

| | SEQUENCE TO | HAS SEEN | DISPOSITION |
|---|---|---|---|
| Paul Thompson | | | |
| Bob Kimmitt | | | |
| John Poindexter | 1 | | |
| Tom Shull | | | |
| Wilma Hall | | | |
| Bud McFarlane | 2 | M | I |
| Bob Kimmitt | | | |
| NSC Secretariat | | | |
| Situation Room | | | |
| JP | 3 | | |

| I = Information | A = Action | R = Retain | D = Dispatch | N = No further Action |
|---|---|---|---|---|

cc: VP Meese ~~Baker~~ Deaver Other _____

COMMENTS JMP Should be seen by: _____
(Date/Time)

Should brief President.

Done 9/25/85.

50

NOT FOR SYSTEM

NATIONAL SECURITY COUNCIL
WASHINGTON, D.C. 20506

~~SECRET/SENSITIVE/EYES ONLY~~

September 23, 1985

INFORMATION

MEMORANDUM FOR ROBERT C. MCFARLANE

THROUGH: JOHN M. POINDEXTER

FROM: JACK MATLOCK

SUBJECT: Possible Deal to Free Shcharansky

I have been informed by State that negotiations for a trade of Shcharansky and a German held by the Soviets (Kraus) for three persons held by the West Germans may be nearing closure. The Germans had offered two persons in the trade and the Soviets asked for a third. The Germans have agreed to a third in principle, but are offering a different person from the one the Soviets requested. Vogel expects to have an answer tomorrow.

If the deal is struck, it is likely that the prisoners will be exchanged in West Berlin next Monday. (This has a certain plausibility, since Gorbachev may want to get it out of the way before he goes to France.)

State informed me further that Rick Burt had recommended that, when and if Shcharansky is released, he have him brought to his residence in Berlin where he could receive a telephone call from the President, and have a photo op. Advance arrangements would also be made to bring Avital to Berlin to meet him.

I think it is a bad idea to involve the President directly, since it may make it harder to solve some of the other human rights cases, particularly since it is the Germans who are providing the trading material. (I have no objection, of course, to bringing Avital to meet him, if and when we are sure it will come off.)

I doubt that State will support Rick's idea of the telephone call, but in case they do, I wanted you to be aware of the situation.

I agree

~~SECRET/SENSITIVE/EYES ONLY~~
Declassify on: OADR