Deadly Round-Off Error
Failure of the Patriot System in Dhahran 1991

Timur Sağlam

Advisor: Georg Hinkel

Abstract On February 25, 1991, a Patriot missile defence system operating in Dhahran, Saudi Arabia, failed to engage an incoming Scud missile. The missile struck U.S. Army barracks killing 28 soldiers and injuring 98. The reason for the failure of the Patriot was a fixed-point round-off error in the range-gate algorithm of the Patriot radar unit's tracking system. This paper reconstructs the events and explains, how the patriot system works. Then it illustrates in detail how the round-off error developed and how it amplified to a critical inaccuracy. Possible approaches for the prevention of these issues are discussed. These approaches, divided in solutions on the technical and on the organisational side of software engineering, will demonstrate that the incident at Dhahran could have been prevented.

1 Introduction

When the Gulf War started, there was nearly no television coverage of the coalition forces military activities. This lead to the Patriot system being the centrepiece of what the public perceived of the Gulf War: Broadcasts of Patriot batteries engaging Scud missiles in the night sky [Post91]. The Patriot system was seen as the representation of the superior western high-end weapon technology. Even George H. W. Bush stated in the State of Union Message of January 29, 1991:

> "Now, with remarkable technological advances like the Patriot missile, we can defend against ballistic missile attacks aimed at innocent civilians."

After the war, this statement and the trust in the Patriot system, was proved to be wrong. In a U.S. congress hearing before the legislation and national security subcommittee of the committee on government operations chairman John Conyers described it as

> "[...] a story of how we projected what we wanted to believe onto the TV screen" [Cong93].

He followed up by saying

> "We thought the Patriot missile was perfect. We were wrong. [...] Ironically, the more information we have, the less successful the Patriot seems" [Cong93].

Whenever the discussion about the efficiency of the Patriot system started, the events on February 25 shattered the trust in the praised surface-to-air missile system. During that night a Patriot missile defence system operating in Dhahran, Saudi Arabia, failed to engage an incoming Scud missile. The Scud hit U.S. barracks killing 28 and injuring 98 U.S. soldiers. The reason for this was a fixed-point round-off error in the range-gate algorithm of the Patriot's tracking system. This caused the Patriot battery to lose track of the incoming Scud and therefore declaring it as a false positive [Offi92b].

This paper focuses on the issues that lead to the failure at Dhahran and on possible approaches for the prevention of these issues. Section 1 explains what the Patriot system is and how it works. Section 2 gives a brief summary about the events around the incident at Dhahran. The third Section examines the round-off error in the range-gate algorithm in detail. How this could have been prevented is examined in the next two sections by giving different solution approaches. These approaches can be on the technical or on the organisational side of software engineering. Section 5 discusses the question which architecture and what programming could have solved the round-off error or at least minimize the loss in precision to a point where it is not a problem any more. Section 6 however discusses what methods and processes could have led to an early detection of the round-off error and then to its fast solution.

2 MIM-104 Patriot System

The MIM-104 Patriot system is a mobile surface-to-air missile system for air defence. It was designed in 1969 and first produced in 1976 by U.S. defence contractor Raytheon. Originally it has been designed to defend against aircraft and cruise missiles. Later, it was updated to deal with the threat of tactical ballistic missiles. This update came in two phases. In the PAC-1 phase the guiding software for the radar and the missiles was changed. It gave the Patriot missiles the ability to intercept tactical ballistic missiles and knock them off course. In the PAC-2 phase the Patriot missiles themselves were improved to make them more powerful. Now they were able to destroy the warheads of tactical ballistic missiles. PAC-1 was used in flight tests for the first time in 1986, PAC-2 in 1987 [Hugh91b]. When the Gulf War started, the U.S. Army was short on PAC-2 missiles. According to the General Accounting Office there were only three PAC-2 missiles in the Army's inventory at the time of the Iraqi invasion [Offi92c].

2.1 The Patriot Setup

The Patriot system operates as a so-called battalion. Each battalion normally consists of six Patriot batteries and a command centre. The command centre coordinates all the batteries. A Patriot battery has six missile launchers, a radar unit, an Engagement Control Station and a Communications Relay Group. A launcher has four containers attached, carrying one Patriot missile each. The Patriot missiles hold warheads weighing around 70 to 80 kilograms. Half of that

weight is fragments. When a Patriot missile comes close enough to a target the warhead detonates and the fragments are shot in the direction of the target to destroy the warhead. Half of the missile weight is propellant, which accelerates the missile to speeds over Mach 5. The Engagement Control Stations from different batteries can communicate with each other. The personnel can define specific areas the Patriot will defend and areas that will not be defended. The Engagement Control Station supports automatic and manual control. In automatic mode the Patriot will use the areas rules to start automatic engagements. For example, if an incoming Scud is predicted to land in the water (or other territory marked as undefended area), the system will not engage that missile [Hugh91b]. The central part of the system is the weapon control computer, whose computer architecture is based on a 1970s design. It performs all major functions for tracking, intercepting and other tasks [Offi92b].

The radar unit has a multifunctional phased array radar that allows to detect, track and illuminate targets at the same time. About 100 targets can be tracked simultaneously and nine concurrent target engagements can be managed [Post91]. It provides ±60-degree coverage in azimuth and can shift the radar beam in 12 μs. This array is used for search, mono pulse tracking and missile guidance tracking. Additionally, the radar unit has a tracking-via-missile receive antenna, which is used to communicate with the Patriot missile during its flight [Schl86]. Another important part of the radar unit is the range gate. It is an electronic detection device in the radar system, which filters all information that does not come from a certain distance range. Only the information from within the distance range gets processed [Offi92b].

2.2 Scud Interception

In the Gulf War the Patriot system was used to intercept a certain type of short-range tactical ballistic missiles, the Scud missile. More specifically, the Iraqis used modified SS-1C Scud B missiles, called Al-Husayn missiles. The modification was extending the centrepiece of missile by welding centrepieces from other Scuds to it. These extensions increased the range of the Al-Husayn missiles so they could reach Tehran from Iraqi territory [Hugh91b]. This modification often leads to the Al-Husayn breaking up at altitudes of 15-20 kilometres. Probably because of this, Raytheon modified the Patriot Software so the interceptor missiles would pursue the faster falling pieces containing the warhead [Post91].

While scanning for targets the radar unit processes the whole radar beam. When the Patriot radar detects an incoming target it checks whether it has the characteristics of a Scud missile. Then the range gate algorithm calculates a range where the target will be next and only processes data from that range. Finding the target in that range confirms that it is a Scud missile. The, now validated, incoming Scud will then be tracked by the range gate algorithm using the same calculation as before [Offi92b]. The Patriot system now calculates the intercept point with the speed, velocity and position of the Scud missile as well as the interceptor's ability to accelerate and manoeuvre. Next the Patriot launcher launches an interceptor on its calculated trajectory. During the

interception flight, commands are sent from the ground via the radar unit to the interceptor to adjust its trajectory if needed. One example for such an adjustment would be if the target does unpredicted changes in its flight path. Additionally, the radar unit illuminates the target with its phased array radar. The interceptor then uses the radar waves reflected from the target to home in on it. Often there was more than one interceptor fired at one target [Post91]. The norm of engaging a Scud missile was launching two interceptors per incoming Scud, because it increases the chance to destroy the target [Hugh91a].

3 Failure at Dhahran

On February 11, 1991, the Patriot Project Office received information from the Israeli military about a 20 per cent shift in the systems range gate after it has been running for eight hours without restarting. This shift was significant, because it meant that the target is not in the centre of the range gate any more, reducing the chance to track the target. On February 16, 1991, a software version was released which improved the accuracy and allowed the Patriot system to run longer without problems. On February 21, 1991, Patriot users were notified about the problem of the range gate shift. They also were told that a software update was on the way. But they were not told how long the system could run continuously without creating a shift that was big enough to affect the precision significantly. According to the U.S. Army the reason for this was that they just falsely presumed batteries do not run their systems long enough to create a problematic inaccuracy. Presumably they never tried to get real data about the run time of the deployed batteries [Offi92b].

On February 25, 1991, the last day of the Gulf War where large military actions occurred, a Scud missile struck U.S. Army barracks of the 14th Quartermaster Detachment out of Greensburg, Pennsylvania, killing 28 soldiers. 98 soldiers were injured, half of them seriously. This incident caused more combat casualties than any other in the Gulf War [Offi92b]. The majority of the soldiers in the barracks had just arrived and were not even completely processed into their units. Helicopters finally evacuated 70 to 100 soldiers to six hospitals, including five Saudi-Arabian facilities [Rost00]. The report of the U.S. General Accounting Office stated that six Patriot batteries protected the airfields of Dhahran. The alpha battery detected the Scud, but could not track it to confirm that it was indeed an incoming Scud missile. As a consequence of this the Patriot battery declared it as a false positive and did not engage. The operators where not shown any sign of the Scud, although they were expecting the missile, because a unit in front of them had tracked the missile as it passed them on its flight to Dhahran.

Although they tried to distribute the software update from the United States to the Patriot locations, the new software version never reached Dhahran in time. The software cassettes containing the update left the MacGuire Air Force Base in New York on February 23 and arrived in Riyadh on February 24, but they were not accorded the highest delivery priority. On the day after the incident, February 26, the software update reached Dhahran [Hugh91d]. According to the

U.S. Army the delay was due to the time they needed to arrange the air and ground transportation of the update into a wartime environment [Post91]. On the same day, the retreat of the Iraqi troops from Kuwait began after they set the oil fields of Kuwait on fire. Two days later, on February 28, President George H. W. Bush declared that Kuwait has been liberated and a ceasefire was in place. This marked the end of the Gulf War.

The 11th Air Defence Artillery Brigade and the Army's Patriot Program Office later investigated the incident. The investigation was not easy because the Patriot battery in question collected no hard technical data. In general no Patriot System had an embedded data recorder [Hugh91d]. During the Gulf War the U.S. Army had 14 portable data recorders for the 26 Patriot batteries in Saudi Arabia and Israel. Many of these recorders were installed several weeks into the war or not at all. A factor that played into that lack of documentation of data might have been the confusion and work load limits on the U.S. engineers that were suddenly and unplanned transferred to Israel during the early days of the Gulf War. Even more important, in Saudi-Arabia, U.S. commanders did not allow the use of the data recorders [Post91].

Later, the Chairman of Subcommittee on Investigations and Oversight of the Committee on Science, Space and Technology of the House of Representatives requested the U.S. General Accounting Office to review the incident as well. This review was, among others, discussed in the U.S. Congress on the hearing about the Performance of the Patriot missile in the Gulf War before the Legislation and National Security Subcommittee of the Committee on Government Operations [Offi92b].

4 Fixed-Point Round-Off Error

What did exactly happen in the Patriots weapon control computer? Why did the radar unit of the Patriot battery at Dhahran fail to track the incoming Scud missile despite it being spotted by a unit in front of the battery as it passed them on its flight to Dhahran? To answer the question, this Section of the paper first gives a short explanation of the internal procedures that lead to the failure at Dhahran. Next it describes how the range gate calculation works, which produced a range gate shift. The third point is about the clock conversion that produced the inaccuracy with its internal computer arithmetic. Next the amplification of the inaccuracy through the computations is explained. At the end the state of the source code is discussed.

The short answer to the question described previously is the following: A fixed-point round-off error in the weapon control computer lead to the radar unit losing tracking of the target. The way the range gate algorithm calculates the position of the tracked missile in combination with the hardware limits of the weapon control computer generated the shift in the systems range gate that was detected by the Israeli military. When the shift is large enough an incoming target will not be found in the range gate, which means the target failed its validation and will be treated as a false positive. This would be a correct

classification, if the range gate calculations were correct. But with the wrong range gate calculations the radar unit was looking for the real target at a wrong position.

4.1 Range Gate Arithmetic

As described previously (see Section 2.2), if an incoming target detected by the radar has the characteristics of a Scud missile, the range gate calculates a range where the target will be next. Then it only processes data from that range of the radar beam. These calculations are made with the range gate algorithm by the weapon control computer. The algorithm stores the targets velocity (speed and direction), latitude, longitude, azimuth and altitude. The next position of the target will be calculated with its last tracked position and velocity. The velocity is stored as an integer and a decimal, for example 3750.2563 miles per hour. Time is defined internally in a clock register as an integer. The time has to be converted to a real number for the calculation, because both the time and the velocity have to be a real number. The conversion is limited through the 24-bit registers and the used computer arithmetic of the weapon control computer. This conversion loss leads to an imprecise calculation of the range gate [Offi92b].

4.2 Clock Conversion

To understand how the conversion loss occurred, it is important to know the difference between floating-point and fixed-point representation. Fixed-point representation is a real data type, where the number of bits for the decimal and the integral parts are fixed. The radix point separates those two parts. The fixed-point representation is depicted in equation 1. In this representation the maximal accuracy for the decimal part does not change. Floating-point representation uses a significand, a base, an exponent and a sign. Each number is represented as equation 2 depicts. Although the bit size of each part is fixed, the precision of the number represented varies depending on the exponent and base [Parh99].

$$x_{fixed\text{-}point} = integral.decimal \tag{1}$$

$$x_{floating\text{-}point} = \pm significand \times base^{exponent} \tag{2}$$

It was the internal computer arithmetic that really produced the inaccuracy. The time was stored in an internal clock as an integer. The clock starts with zero on the system start and then measures time in tenth of seconds. To convert it to a real number, it was multiplied by 0.1_{10} in binary fixed-point representation stored in a 24-bit register. The reason for the conversion is that they needed the time in seconds and the clock value in tenths of seconds. The problem is that the base two representation of 0.1_{10} is non-terminating because each n_{th} bit of the decimal part in the register represents 2^{-n}. Therefore a decimal of 0.001_2 is 0.125_{10} and a decimal of 0.0001_2 is 0.0625_{10}. So the 0.1_{10} was basically $0.1 \times (1 - 2^{-20})$, which was represented as $0.000110011001100110011001100_2$ in the 24-bit register. The newly calculated clock value in seconds was then stored in a pair of 24-bit

registers, presumably using fixed-point representation again. Next the time value in the two registers was transformed into a 48-bit floating-point number whose decimal accuracy was limited to 24 bit due to its transformation source. The time difference between two radar pulses, including their inaccuracy, was then used to calculate the new position of the target with its velocity [Skee92,Parh99].

4.3 Run Time Amplification

What increased the inaccuracy to a point where it became a problem? According to the report of the General Accounting Office the effect of the inaccuracy is linear proportional to the targets velocity and the Patriot systems runtime. A Scud flies approximately at 3750 miles per hour. Such high speeds are significantly increasing the effect of the inaccuracy. But the runtime is an even bigger factor in this equation because the higher the initial clock in tenths of seconds is, the greater the round-off error gets. The Israeli military measured a 20 per cent shift of the range gate after the system has been running for eight hours at a time. These 20 per cent were 55 meter in total and equalled a clock error of 0.0275 seconds. According to Patriot Project Office officials the Patriot system will not track a Scud when there is a range gate shift of 50 per cent. This shift was calculated to appear after 20 hours of continuous use. But Army officials

Table 1. Effect of the Runtime on the Inaccuracy, taken from [Offi92b]

Time (Hours)	Time (Seconds)	Calculated Time (Seconds)	Inaccuracy (Seconds)	Range Gate Shift (Meters)
0	0	0	0	0
1	3600	3599.9966	0.0034	7
8	28800	28799.9725	0.0275	55
20	72000	71999.9313	0.0687	137
48	172800	172799.8352	0.1648	330
72	259200	172799.8352	0.2472	494
100	360000	355999.6667	0.3433	687

believed that users were not running their Patriot system for longer than eight continuous hours. But the battery in question at Dhahran was running over 100 hours on February 25, 1991. This runtime in seconds ($100h * 60 * 60 * 10$) equals a clock value of 3600000_{10} or 11011011011101000000_2, an error of 0.3433 seconds and a range gate shift of 687 meters. Those 687 meters are a little more than a 250 per cent range gate shift, which is five times higher than the shift where the Patriot cannot track a Scud any more. See table 1 for more details on the precision loss [Offi92b].

4.4 State of the Source Code

The source code of the weapon control computer consisted of over one million lines of code [Hugh91b]. Software modifications were often implemented under time pressure. For example, the software modification that gave the Patriot system the ability to intercept Scuds was assessed and incorporated in less than one week by the U.S. Army and Raytheon in 1990 [Cong93]. The report of the General Accounting Office states that the software of the Patriot system was modified six times during the Gulf War. At least one of these changes added a new subroutine for converting the clock value (in tenths of seconds) more precise into floating-point representation. This subroutine was needed in about six points of the program, but was not inserted in every one of them. As a consequence the calculation of the time difference got less precise because the precision error was not cancelling itself when two time values with different precision were subtracted from each other [Skee92]. The state of the source code with its errors, described in this section, contributed to the failure at Dhahran by further increasing the inaccuracy.

5 Technical Prevention

This Section tries to give technical approaches for the correction of the fixed-point round-off error. These three solutions probably would have prevented the incident at Dhahran by increasing the precision of the system to a point where the radar unit would have been able to successfully track the scud missile, enabling the Patriot to launch an interceptor. The first approach introduces the idea of a forced system restart. The next approach suggests increasing the register size of the weapon control computer. The third and last approach tries to avoid the clock conversion itself. The Section discusses each approach, evaluates how efficient they could have been and considers combinations of those three approaches.

5.1 Forced System Restart

Maybe the simplest approach would be a forced restart of the weapons control computer. Because the internal clock has a drastic effect on the radar unit's inaccuracy, a restart would reset the inaccuracy to zero (see table 1). The system would cycle through a loop with growing inaccuracy from zero up to the point where the reset occurs. This would limit the maximal inaccuracy. The biggest factor for the approach of a forced restart is an appropriate time frame for the restart itself. Since a Scud cannot be tracked if the range gate shift is greater than fifty per cent, which is reached at 20 hours of continuous use, the forced restart could happen at a run time of 15 hours. The operators could be warned two hours in advance and restart the system early if they assess the point of forced restart as problematic timing wise. At 15 hours the range gate would be shifted by approximately 37,5 per cent. This time frame for the forced restart is a simple suggestion from a computer science view on the papers sources, and in

no way supported by military knowledge. A practical time frame would have to be determined by a team of military and technical experts.

It is important to note that even after 10 hours there will be a range gate shift by approximately 25 per cent. Before the solution of a forced restart can be implemented, extensive tests about the effects of the range gate shift have to be conducted. Especially the question whether a shift under 40 per cent (or under a lower percentage for different time frames) has drastic effects on the Patriots performance has to be answered. The reason for this is that this solution accepts a growing inaccuracy over time to a certain extend, until the system restart resets the clock. If for example a range gate shift between 30 and 50 per cent would still affect the Patriots performance, a time frame of 12 hours (and a warning at 10 hours) until a forced restart would be more appropriate. To reduce the range gate shift even more, for example a restart every eight hours is necessary. Because the Patriot batteries operate in battalions and, according to the General Accounting Office [Offi92b], the restart of a Patriot battery takes 60 to 90 seconds, the dangers of a restart should not be that high. The battalions could coordinate their time frames to guarantee that only one Patriot battery restarts at a time.

But how good is this solution of a forced restart? First off, this approach does not fix the inaccuracy itself. It just resets the amplification factor of the inaccuracy, which increases over time. The advantage of this approach is the simplicity of its implementation. But the downside of this approach is that there are regularly short time periods where the Patriot system is not operational. It would be a good idea to combine this approach with another solution that improves the inaccuracy itself (see Section 5.2). If the range gate shift increase is lower, a larger time frame can be chosen, which reduces the percentage of the system's downtime and hence the significance of this disadvantage.

5.2 Increased Register Size

A more elegant approach would be to use larger registers for the essential calculations of the range gate algorithm. The Patriot system used 24-bit registers in its computer architecture. These types of registers are too small for accurate calculations from a modern standpoint. A 64-bit architecture would significantly increase the precision. For example a fixed point representation in a 64-bit register with the same amount of integral bits, which means the same amount of bits before the radix point, would have 52 bits for the decimal part, which is 40 bits more than the 24-bit representation has. This means that in that case the 64-bit representation has 40 more binary digits for the decimal part of the real number. Floating-point numbers would also be more precise, depending on the sizes of the exponent and the base (see Section 4).

There are three parts of the range gate algorithms source code that are important for this solution. In each of them a larger register would be needed for more precision in the range gate calculation. The first one is the register that contains one-tenth. It is the fixed-point register that is used to convert the clock value in tenths of seconds to a value in seconds by multiplying the one-tenth with

the clock value in tenths of seconds. As stated before in Section 4, 0.1_{10} cannot be represented in fixed-point arithmetic because the base two representation of 0.1_{10} is non-terminating. Increasing the size of this register would make the conversion of the clock more precise. The second one is the register where the converted clock time is saved in fixed-point representation. Because of the conversion with one-tenth the clock time, which was originally a whole number, is now a real number. This means that no matter how precise 0.1_{10} is represented, the result only can be accurate if the result register has enough bits so it does not cut off a part of the decimal value. The third one is the register were the time values are stored after their conversion from fixed-point to floating point. The reason is the same as before: No initial precision matters if a result register at the end cannot represent a number that precise.

This approach minimizes the inaccuracy, which is still amplified by clock time (see Section 4.3). But using registers with enough bits can reduce the inaccuracy to a point where the system can run much longer continuously without shifting the range gate to a problematic point. It can be combined with the previous attempt if the increase of the register size is not enough. Raytheon made several updates, where at least one of them tried to implement this solution. But according to Robert Skeel they did not insert the subroutine, which used higher precision, to every line in the source code where it was needed [Skee92]. This second approach is very effective, because it can greatly reduce the range gate shift. The disadvantage is that it can only be easily implemented to a certain bit size. The hardware the system was running on certainly had its limits regarding that matter. A higher bit size would need to update the hardware, which is way more complex than only changing the software. New hardware might mean different specifications that could lead to a lot more software changes.

5.3 Avoidance of the Clock Conversion

A third approach would be to avoid the one-tenth multiplication all along. The Problem with the 0.1_{10} in binary fixed-point representation stored in a 24-bit register is not the multiplication for the conversion itself alone. At least as important is that this error gets amplified through the run time of the clock. As already described in Section 4.3, the longer the system runs continuously, the worse the error gets. This is important for this approach, because fixing the problem with one-tenth multiplication also fixes the problem with the amplification. To sum it up it can be explained as followed: If there is no run time clock inaccuracy in the first place, there is nothing to amplify.

One way to implement this approach would be to use a clock which increments every second instead of every tenth of a second The problem is that the engineers who designed the weapon control computer probably had good reasons to use a clock in tenth of seconds. That means this implementation of the approach is probably a very naive suggestion, but we cannot know that without full knowledge about the Patriots internal procedures and algorithms. A better way to implement this approach of avoiding the one-tenth multiplication is to use the clock value in tenths of seconds. This means further calculations have to adapt

their arithmetic procedures to these circumstances. For example when the next position of Scud is calculated, its velocity gets multiplied with the time passed since the last radar pulse. If we use the original clock in tenth of seconds, the time difference is also in tenths of seconds. The time difference could originally be multiplied with the velocity in miles per seconds. This velocity now needs a conversion to miles per tenths of second.

A valid concern is that we now have a one-tenth multiplication at another point of the algorithm. We removed the conversion of the clock and added a conversion of the velocity. How valid this concern is, depends on how the velocity is measured and stored. A report of the General Accounting Office [Offi92b] indicates that the velocity is stored in miles per hour as a real number. That means that the velocity already has to be converted if we use, like the system originally did, a time difference in seconds. A conversion would either transform to miles per second or meters per second. So there probably already is a small conversion loss of accuracy because there is already a conversion: The conversion of the velocity. But even if this conversion had been originally accurate and therefore there had not been a conversion loss from miles per hour to miles per second (or meters per second), we still would have improved the systems accuracy even though we just moved the conversion to another point. The reason for this is that we moved the conversion error to a number that is in a fixed range. A Scud missile has velocity around Mach 5, which means it never gets faster than 4500 miles per hour. Even if this velocity is stored as meters per second the value, 2011.68 meters per second, is smaller than the time values that were multiplied by one-tenth (see table 1). That means the inaccuracy stays small enough to not be a problem. To summarise we can say we moved the conversion to a place in the algorithm where it is not amplified by the clock (see Section 4.3) and therefore not a problem for the accuracy.

Now how good is this third approach? It probably has the biggest effect of all three. It also removes the need for the approach of the forced restart (see Section 5.1) because it removes the inaccuracy amplification. The third approach can still be combined with the approach using larger registers (see Section 5.2) to further improve the accuracy. The disadvantage of this third approach is that the software changes are not as easy to implement as the ones from the approach of the forced restart because the approach changes the most important calculations of the range gate algorithm. In conclusion, it can be said that if the time and resources are available, the last two approaches are both more elegant and more efficient, but the first one is probably the easiest to implement.

6 Organisational Prevention

Apart from the technical issue there were several issues on the organisational side of software engineering. These issues were not only connected to the incident at Dhahran. They were also present throughout the Patriots deployment during the Gulf War. Theodore A. Postol, a professor at the Massachusetts Institute of Technology, wrote a paper about the lessons of the gulf war experience [Post91]

in which he discussed and criticized the Patriots performance. This paper was later a big part of a hearing of the U.S. Congress about the Patriots performance [Cong93]. Early reports of the U.S. Army during the Gulf War stated that 96 per cent of the Scuds over Israel and Saudi Arabia were engaged successfully. Later, the U.S. General Accounting Office found out only about nine per cent of the engagements in Operation Desert Storm had strong evidence for resulting in a warhead kill [Offi92a].

This Section only focuses on the organisational issues that were directly connected to the incident at Dhahran. For more details on the whole Patriot deployment during Operation Desert Storm see Postol's paper "Lessons of the Gulf War experience with Patriot" [Post91] and Stein's and Postol's paper "Patriot Experience in the Gulf War" [StPo92]. In total the Section discusses four issues connected to Dhahran and their consequences as well as explaining what could have been done to prevent them.

6.1 Inquiry of Data during the Deployment

Maybe the biggest organisational issue was the complete failure to collect a sufficient amount of data during the deployment. The Patriot system itself was not equipped with an embedded data recorder of any kind. The only way to obtain data from a Patriot battery was to use an external, portable data recorder. But the U.S. Army only had 14 of these portable data recorders. During the Gulf War there were 20 Patriot batteries deployed in Saudi Arabia and 6 in Israel. That means in a best-case scenario only about half of the batteries could have been equipped with a portable data recorder. But that was not the case. As touched on in Section 3, many of these recorders where installed several weeks into the war or never at all [Offi92b]. Problems were the workload limits and the confusion of the U.S. engineers that were suddenly and spontaneously transferred to Israel during the early days of the Gulf War. When they arrived in Israel the engineers and soldiers of the Patriot batteries had serious system problems they had to fix, so there was little time to worry about data recorders. Embedded data recorders would have solved that problem because they would have neither required installation nor any kind of attention at that moment. In Saudi Arabia, there probably would have been enough time to install these portable data recorders. But because of a malfunction that appeared right after a data recorder had been plugged in, U.S. commanders did not allow the installation of these data recorders at all. Also in this situation embedded data recorders would have solved that problem, because they would not have to be installed [Post91].

Even in the rare cases they collected data with these recorders, the data was not sufficient for analysis later on. The Army obtained data about the points in time where the Patriot system detected a target. Also the data gave information whether the detected object matched the speed criteria of the modified Scud missiles. A third fact that the data contained was whether the targets impact point would hit an asset that is protected by the Patriot. If it was the case, an engagement was started. The last thing the data contained was whether the Patriot reported a warhead kill. But that warhead kill was defined by reaching the

calculated intercept point and the radar unit losing contact with the interceptor. This data could not prove that an interceptor accurately hit the target, only that it flew to a specific point and detonated. The data was able to prove whether or not the fuse of the interceptor reacted quickly enough to destroy the Scud. But this information was never processed because the Patriot project officials believed it would not benefit the assessment process [Cong93].

Ironically Israel deployed scientists to observe the Patriot engagements, but initially their results were not taken seriously (see Section 6.4). That shows that the inquiry of a sufficient amount of data during the deployment was definitely possible. It is fair to say that the Patriot Project Office failed to see the importance of the task. This issue is just another one that probably delayed the discovery of the range gate shift. So without this issue, the incident at Dhahran may have been prevented by an early update to the Patriot systems range gate algorithm. Furthermore, this issue would have been non-existent if Raytheon would have designed the system with a factory installed internal data-recording device and the device would be capable to collect comprehensive data about the engagements and their success. Theodore A. Postol already described it as early as in 1991 as an oversight that is difficult to comprehend in this modern age of digital electronics and data storage devices. The today's importance of testing in software engineering increases even more the difficulty to comprehend the oversight of inquiry of data during the deployment.

6.2 Comprehensive Testing of Updates

In the Gulf War, two updates were made for the Patriot's Software. Both involved changes to several hundred lines of code [Hugh91c]. Even though they were made after careful investigation, detailed design and a live firing test on a test range as well as extensive tests in a simulation, they made these changes according to Colonel David Heebner, commander of the U.S. Patriot crews in Israel, in two weeks. Under normal conditions, he said, they would have taken two years [StPo92]. But while trying to solve the problem that lead to the incident at Dhahran (see Section 4), they did not insert the updated subroutine to every line in the source code where it was needed (see Section 5.2). The problem was that because of the on going conflict, there was no time to test the updates enough. Also the engineers that designed the updates had very limited data to work with (see 6.1) [Hugh91c]. They did test the updates in a simulation, but they did not realise that the patches were not a hundred per cent correct.

In this context we can conclude that at least one of the following three assumptions occurred: Either they had software errors in their simulation code, which produced invalid testing results. Or they failed comprehensive testing with the simulation and therefore did not find the errors in the update. A last assumption is that the bugs in the updates could not be detected in a simulation environment. In this case comprehensive Patriot tests would have been required. Either way, this was another of many factors that lead to the incident at Dhahran.

6.3 Explicit Operation Instructions

As discussed before in Section 3, on February 21, 1991, Patriot users were notified about the problem of the range gate shift. They also were told that a software update was on the way. The simple solution until the update would arrive was to restart the system regularly by hand. This is the manual equivalence to the first approach from Section 5.1. This temporary solution could have worked if they had told the Patriots users how long they could run the Patriot system without restarting it manually. The problem was that they did not do that. According to the U.S. Army the reason for this was that they just falsely presumed batteries would not run their systems long enough to create a problematic inaccuracy. They thought a Patriot battery would not run longer than eight hours at a time. Presumably, they never tried to get real data about the run time of the deployed batteries [Offi92b].

But what does that mean? They were responsible of a military system that was already in use in a wartime environment. This system was flawed so it stopped working after a certain run time. And even though this critical flaw was known and its existence was communicated to the users, no one thought of telling the users the actual run time when the system fails on every occasion. From a modern standpoint this behaviour is absurd. A case like this in the private sector would have been bad, but in the military sector, for a system whose purpose is to protect lives, this was catastrophic. Appropriate behaviour in this situation alone could have lead to the prevention of the incident at Dhahran.

So what could have been done? One way would have been to give a comprehensive report to the Patriot's users, including the known data about the run time to range gate shift relation. If the users had known their system fails on every occasion at a run time of 20 hours, they could have restarted their system more often. Another approach would have been to start an investigation about the average run times of the batteries in Israel and Saudi Arabia. That would have lead to the Army command realising they underestimated the normal run time and eventually them notifying the Patriot's users.

6.4 Lack of Cooperation with Israel

One smaller issue was the lack of cooperation with the Israelis. There were early complains about issues with the Patriot system. Raytheon thought that the reason for this was the Israelis not following U.S. fire doctrine. According to Raytheon, this included adjustments to operational procedures and experimentation with the fire control doctrine. Later it was the Israelis who informed the Patriot Project Office about the 20 per cent shift in the systems range gate. They investigated that issue by observing the Patriot engagements with their own scientists and measuring equipment presumably from the Israeli missile test range [Post91].

These issues show that the U.S. Army and Raytheon with the Patriot Project Office were not able to acknowledge issues that were brought up from outside. It could be described as an organisational bias that prevented them from seeing

the flaws of their project. We can only speculate whether without this attitude the range gate shift would have been detected early, and whether the incident at Dhahran could have been prevented, because the updates would have arrived in time. But no matter what, they should have been open to hints about potential flaws of the Patriot system.

7 Conclusion

In the end, the Patriot revealed itself as a flawed system. Its success proved to be a lot smaller then it initially appeared. Only about nine per cent of the engagements in operation Desert Storm could be proved to result in a warhead kill. Additionally, no clear evidence was found for a reduction of the ground damage in Israel. What they found was that Patriot interceptors failed to destroy incoming Scuds in a good number of situations [Post91,Cong93].

The incident at Dhahran was the most tragic of these situations. The fixed-point round-off error in the weapon control computer lead to a shift in the radar units range gate. The Patriot's radar unit in Dhahran looked for the Scud 687 meters from its real location to confirm its existence. And on this position there was no sign of a Scud missile. The system then assumed it was a false positive and did not start an interception. The technical errors themselves quite were simple but had vast consequences: 28 soldiers were killed and 98 soldiers were injured, half of them seriously. A couple technical and organisational issues lead to this incident. For both types of issues the paper has given approaches that could have prevented the incident.

On the technical side the three approaches were given that would have prevented that incident: First a forced system restart to reset the time amplification of range gate shift. Second the approach of the increased register size for higher precision and therefore a lower range gate shift. Third, avoiding the clock conversion to reduce the inaccuracy and remove the amplification. On the other side of software engineering, the organisational side, four issues were addressed and an approach for each of them given: First, the importance of collecting data during the Patriot's deployment. This could have lead to an early discovery of the range gate shift. Second, comprehensive testing of the software updates. The inconsequential use of updated subroutines increased the inaccuracy. Third, explicit instructions regarding the handling of the range gate shift until the update would arrive. Those could have prevented the incident regardless of the range gate shift. Fourth and last, the openness about flaws in their system.

In conclusion there are several lessons that can be learned from that incident. First, one can never be too careful while writing software. It is important to be absolutely sure how precise a system has to be. Second, one can never underestimate the importance of proper testing and the collecting of data in the environment it was made for. Third, one cannot rely on assumptions about user behaviour, specially if your device was developed for saving lives.

References

Cong93. Congress of the United States. House Committee on Government Operations Legislation and National Security Subcommittee. *Performance of the Patriot missile in the Gulf War: hearing before the Legislation and National Security Subcommittee of the Committee on Government Operations, House of Representatives, One Hundred Second Congress, second session, April 7, 1992*. Nr. Bd. 1. U.S. Government Printing Office. 1993.

Hugh91a. David Hughes. Joint U.S.-Israeli Forces Use Patriots To Defend Against Iraqi Scud Missiles. *Aviation Week and Space Technology*, Band January 28, 1991, S. 34.

Hugh91b. David Hughes. Patriot Antimissile Successes Show How Software Upgrades Help Meet New Threats. *Aviation Week and Space Technology*, Band January 28, 1991, S. 26–28.

Hugh91c. David Hughes. Success of Patriot System Shapes Debate on Future Antimissile Weapons. *Aviation Week and Space Technology*, Band April 22, 1991, S. 90–91.

Hugh91d. David Hughes. Tracking Software Error Likely Reason Patriot Battery Failed to Engage Scud. *Aviation Week and Space Technology*, Band June 10, 1991, S. 25–26.

Offi92a. United States General Accounting Office. *Operation Desert Storm: Data Does Not Exist to Conclusively Say How Well Patriot Performed*, Band 1. U.S. Government Printing Office. 1992.

Offi92b. United States General Accounting Office. *Patriot missile defense: software problem led to system failure at Dhahran, Saudi Arabia : report to the Chairman, Subcommittee on Investigations and Oversight, Committee on Science, Space and Technology, House of Representatives*, Band 1. U.S. Government Printing Office. 1992.

Offi92c. United States General Accounting Office. *Project Manager's Assessment of Patriots Missile's Overall Performance is Not Suported*, Band 1. U.S. Government Printing Office. 1992.

Parh99. Behrooz Parhami. *Computer arithmetic: Algorithms and hardware designs*, Band 20. Oxford university press. Department of Electrical and Computer Engineering, University of California, 1999.

Post91. Theodore A. Postol. Lessons of the Gulf War experience with Patriot. *International Security, The MIT Press*, 16(3), 1991, S. 119–171.

Rost00. Bernard Rostker. Iraq's Scud Ballistic Missiles. iraqwatch.org, July 2000.

Schl86. D. C. Schleher. *Introduction to electronic warfare*. The Artech House radar library. Artech House, Dedham, MA. 1986.

Skee92. Robert Skeel. Roundoff Error and the Patriot Missile. *SIAM News*, 25(4), 1992, S. 11.

StPo92. Robert M. Stein und Theodore A. Postol. Patriot Experience in the Gulf War. *International Security, The MIT Press*, 17(1), 1992, S. 199–240.